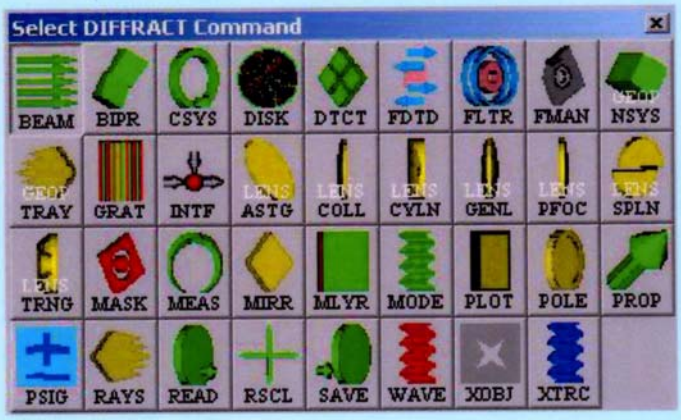
DiffractGUI™

User's Guide

Graphical User Interface

for

DIFFRACT™ 7.7, 7.9, 8.1, 8.2, 8.3, 8.4



A.V. Cazan

January 2007

日本語訳

目次

1. DIFFRACT の Graphic User Interface(GUI)によるシミュレーションのセットアップと実行についての説明（Instructions to set up and run simple simulations within Diffract's Graphic User interface (GUI)・・・・・7
2. DiffractGUI のインストール（Installing DiffractGUI）・・・・・19
3. 序論（Introduction）・・・・・21
   - 3.1 モデルファイル（Model Files）・・・・・21
   - 3.2 プログラムの起動（Start Up）・・・・・23
   - 3.3 編集バー（Edit Bar）・・・・・24
   - 3.4 ツールバー（Tool Bar）・・・・・25
   - 3.5 長さの単位の取り扱い（Handling length units）・・・・・26
   - 3.6 プログラムの構成（Configuration）・・・・・27
   - 3.7 フォント（Fonts）・・・・・28
4. Logfile の観察（The loggerview）・・・・・28
   - 4.1 LoggerView の option (LoggerView options)・・・・・29
5. Workbench・・・・・30
   - 5.1 object のチェック（Control Elements）・・・・・30
   - 5.2 Workbench の options (Options for workbench)・・・・・31
   - 5.3 新しい objct の導入（Create a new object）・・・・・32
     - 5.3.1 ChooseObject 会話ボックス（ChooseObject dialogue）・・・・・32
     - 5.3.2 Command Panel ・・・・・32
   - 5.4 Object の削除（Delete an object）・・・・・33
   - 5.5 切り取り、コピー、貼り付け（Cut, copy, paste）・・・・・33
   - 5.6 Object の順序の変更（Change the order of object）・・・・・34
   - 5.7 Object の DIFFRACT パラメーターの変更（Change DIFFRACT parameters of an object）・・・・・34
   - 5.8 $命令エディター(The $commands editor)・・・・・39
6. Script の編集（ScriptView）・・・・・41
   - 6.1 ScriptWizard ・・・・・43
   - 6.2 DIFFRACT menu command block の新設（Creating a new DIFFRACT menu command block）・・・・・43
     - 6.2.1 Using the command panel ・・・・・43
     - 6.2.2 キーボードからのタイピング（Typing from the keyboard）・・・・・45
   - 6.3 DIFFRACT menu の command block の削除(Deleting a DIFFRACT menu

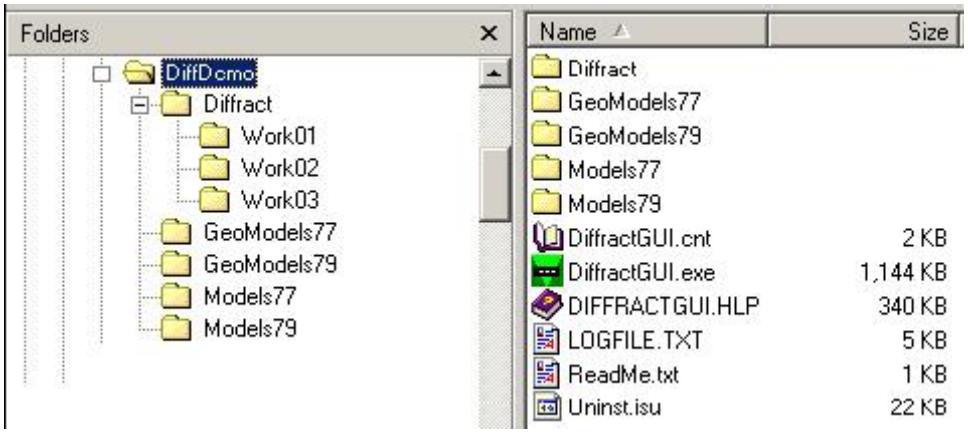
command block) ・・・・・46
6.4 DIFFRACT menu command block のパラメーター値の変更 (Changing parameter values of DIFFRACT menu command blocks) ・・・・・48
6.5 $命令（$commands)・・・・・49
7. GeoSysView ・・・・・51
7.1 方向とズーム（Orientation and Zoom)・・・・・52
7.2 Scene Option ・・・・・53
7.3 GeoSysView Options ・・・・・54
7.4 新しい光学面の生成（Creating a new surface)・・・・・54
7.4.1 ChooseModel dialog の使用（Using ChooseModel dialog) ・・・・・55
7.4.2 SurfacePanel の使用（Using SurfacePanel) ・・・・・55
7.5 光学面の削除（Deleting a surface) ・・・・・56
7.6 光学面の順序の変更（Changing the order of surfaces) ・・・・・56
7.7 光学面のパラメーターの変更（Changing the order of surfaces) ・・・・・57
7.7.1 Edit DIFFRACT parameters dialog の使用（Using Edit DIFFRACT parameters) ・・・・・57
7.7.2 キーボードの使用（Using the keyboard) ・・・・・57
7.8 光線追跡（Ray tracing)・・・・・57
7.8.1 Initial Ray の変更（Changing the Initial Rays) ・・・・・57
7.8.2 厚さの dialog box(Thickness Dialog Box) ・・・・・62
7.8.3 光学面との交差（Hits on Surfaces) ・・・・・63
7.9 幾何光学系の ZY 断面の表示（View the ZY-section of a geometric-optical system)・・・・・66
7.10 外部ファイルによる GEOP 光学面の導入(Importing GEOP Surfaces from External Files) ・・・・・67
7.10.1 ファイルの文法(File syntax) ・・・・・67
7.10.2 例 ・・・・・70
7.10.3 Zemax(*.zax)ファイルの導入（Input Zeemax (*.zax) files)・・・・73
7.11 マウスホイールによる光学面のパラメーターの変更（Modify Surface Parameters woth the mouse wheel)・・・・・ 76
8. SignalView ・・・・・79
8.1 Signal Data の取り扱い（Handling Signal Data)・・・・・80
8.2 カーブの変更（Changing the Curves)・・・・・83
8.3 ディスプレイの変更（Changing the Display)・・・・・84
8.4 Zooming and Panning ・・・・・85
8.5 Diplay Configuration の保存と読み込み (Save and Load the Display

Configuration) ・・・・・85
9. BitmapView ・・・・・87
10. ReliefView ・・・・・101
10.1 ReliefView のためのファイルの読み込み(Loading a file for ReliefView)・100
10. 2 表示の option（Scene options)・・・・・102
10.3 方向とズーム（Orientationand Zoom)・・・・・104
10.4 データ値の収拾（Picking data values)・・・・・105
11. Session View ・・・・・109
12. DIFFRACT の起動 ・・・・・111
12.1 DIFFRACT 起動のための option (Options for DIFFRACT launcher) ・・112
13. Working Directory の除去（Cleaning Up Working Directory)・・・・・115
14. Zernike 収差係数の Import/Export（Import/Export of Zernike Coefficients)・・117
14.1 ファイル形式（The File Format)・・・・・117
14.2 A_{nm} と B_{nm} の単位（Units for A_{nm} and B_{nm})・・・・・123
15. 位相/振幅 Object として使われる Mask Data Files（Mask Data-files used as Phase/Amplitude Objects)・・・・・124
15.1 WYKO ファイルフォーマット(+.asc, .sdf)で保存された分布の導入(Importing distributions stored in WYKO file format(.asc, *.sdf)) ・・・・・139
15.2 WYKO *.asc ファイルの望ましい形式 (Expected format of the WYKO.asc files) ・・・・140
15.3 データを DIFFRACT 分布ファイルとして保存 (Store the data as DIFFRACT distribution file(*.dst) ・・・・・142
15.4 ユーザーが定義した開口（User defined apertures)・・・・・142
15.5 2π の位相不連続を解消するための Phase-unwrapping（Phase- unwrapping to remove 2π phase discontinuities)・・・・・144
15.6 データアレイのサイズ変更(Resizing the data arrays)・・・・・145
15.7 位相マップから Zernike 係数の抽出（Extraction of Zernike coefficients from phase distribution)・・・・・147
15.8 MaskEditor 内の算術演算（Arithmetic Parser Inside the MaskEditor)・・154
15.9 MaskEditor の scripts (Scripts in the Mask-Editor)・・・・・154
15.10 Bitmap としての単純な mask-plot の導出（Exporting Single Mask-Plots as Bitmaps)・・・・・158
15.11 ASCII データマトリクスとして単純な mask-Plot の導出（Exporting Single Mask-Plots as ASCII data-matrices)・・・・・158
15.12 Bitmaps からのデータの導入（Importing Data from Bitmaps)・・・・・159
15.13 任意の ASCII ファイルからのデータ導入（Importing Data from

Custom ASCII Files）・・・・・164
16. 座標の解析（Coordinate Analyser）・・・・・177
16.1 Mirror ・・・・・178
16.2 多層膜（Multilayer）・・・・・181
17. ディスクの編集（Disk Editor）・・・・・184
17.1 序論・・・・・184
17.2 ディスクエディターの起動（Launching the DiskEditor）・・・・・185
17.3 The Main DiskEditor Window・・・・・186
17.4 ディスク表面の構造（Disk Surface Structure）・・・・・191
17.5 画像による構造の編集（Edit Structures Graphically）・・・・・195
17.6 垂直断面（Vertical Cross Section）・・・・・198
17.7 3 次元表示（3D View）・・・・・199
18. ビームエディター（Beam Editor）・・・・・202
18.1 序論（Introduction）・・・・・202
18.2 ビームエディターの起動（Launching the BeamEditor）・・・・・203
18.3 BeamEditor のメインウインドウ（The Main BeamEditor Window）・・・205
18.4 BEAM パラメーター（BEAM parameters）・・・・・209
18.5 パラメーターの図表的編集（Edit Parameters Graphically）・・・・・212
18.6 垂直断面（Vertical Cross Section）・・・・・214
18.7 3 次元表示（3D View）・・・・・215
19. ワークフロー編集（Workflow Editor）・・・・・216
19.1 Script Editor(The Script Editor)・・・・・219
19.2 Workflow script の言語（The Workflow Scripting Language）・・・・・221
19.2.1 変数と定数（Variables and Constants）・・・・・222
19.2.2 表現、関数および記述（Expressions, Functions and Statements）・223
19.2.3 Control Structures・・・・・224
19.3 Tutorial・・・・・226

1. DIFFRACT の Graphic User Interface(GUI)によるシミュレーションのセットアップと実行についての説明

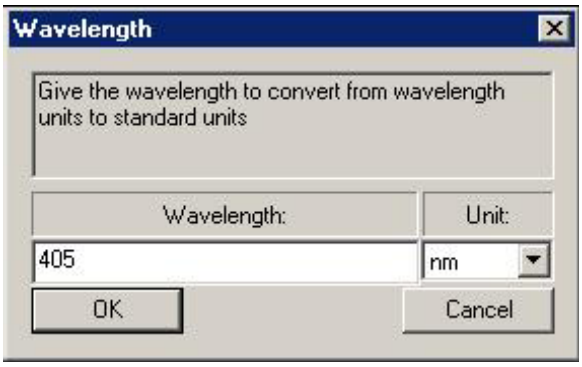
本章の説明は DIFFRACT の Graphic User Interface(GUI)をコンピューターにインストールし、いくつかの簡単なシミュレーションを実行するために役立つことと思う。はじめに、**menu バー**に並んであるボタンになれておく必要がある。まずカーソルをそれぞれのボタンの上に置くと、間もなくそのボタンの名前がスクリーンに表示される。特定のボタンについてさらに詳しい説明がほしいときは、まず **Help** ボタンをクリックする。疑問符（？）がカーソルのそばに現れるので、（？）がカーソルのそばにあるうちに、カーソルを希望のボタンの上に drag し、クリックすればよい。すると、対応する help ファイルが開かれ、必要な情報を見ることができる。



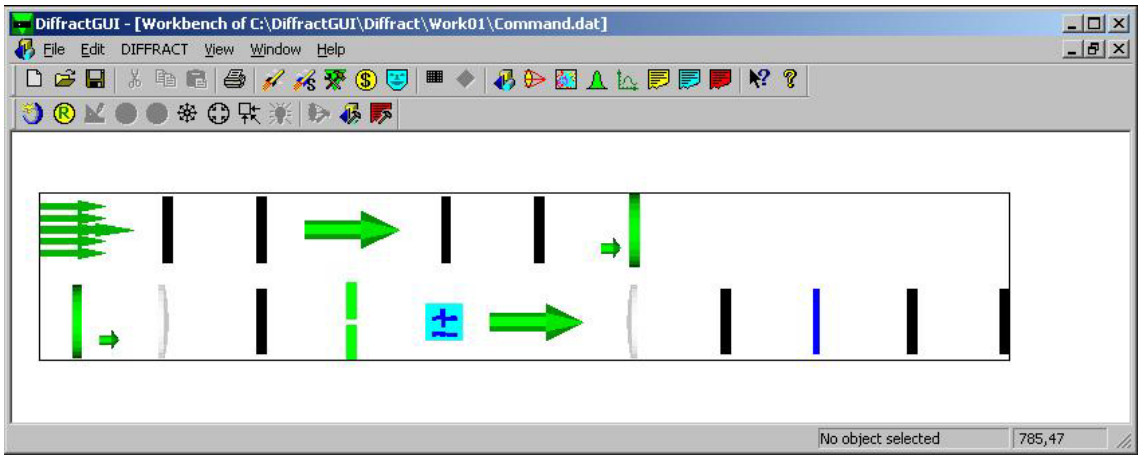
1. プログラムディスクの SETUP.EXE を実行し、ハードディスク上にディレクトリ、たとえば C:¥DiffractGUI を指定する。インストールが実行されると、C:¥DiffractGUI にはいくつかのファイルとサブディレクトリ（またはフォルダー）が作られる。C:¥DiffractGUI の中のフォルダーの一つは**<DIFFRACT>**である。C:¥DiffractGUI¥Diffract の中には **Diffract.exe**, **Settings.log** といくつかの作業サブディレクトリ**<Work01>**, **<Work02>**, **<Work03>**などが存在している。（註：**Diffract.exe** をダブルクリックすれば、DIFFRACT Ver.8.2 が Graphic Unse Interface なしで起動する。したがって、このプログラムを起動しないで、その代わりにホームディレクトリ C:¥DiffractGUI の **DiffractGUI.exe** を起動することをお勧めする。）次に示す手筈によって、Diffract とその GUI を Command.dat ファイルから step-by-step 法で実行することができる。一旦この test run に慣れれば、どのようなシミュレーションも DIFFRACT とその GUI で実行することができる。

2. C:¥DiffractGUI ディレクトリに戻り、DiffractGUI.exe をダブルクリックする。タイトルスクリーンの次に、DiffractGUI のメインウインドウが現れる。Menu バーから File_Open を指定し、C:¥DiffractGUI¥Diffract¥Work01¥Command.dat を選んで、その

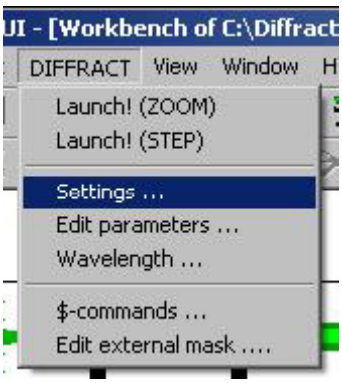
ファイルネームをダブルクリックする。会話ボックスが開かれ、このシミュレーションのために選ばれた波長（cm, mm, μm, または nm の単位で）を表示する。

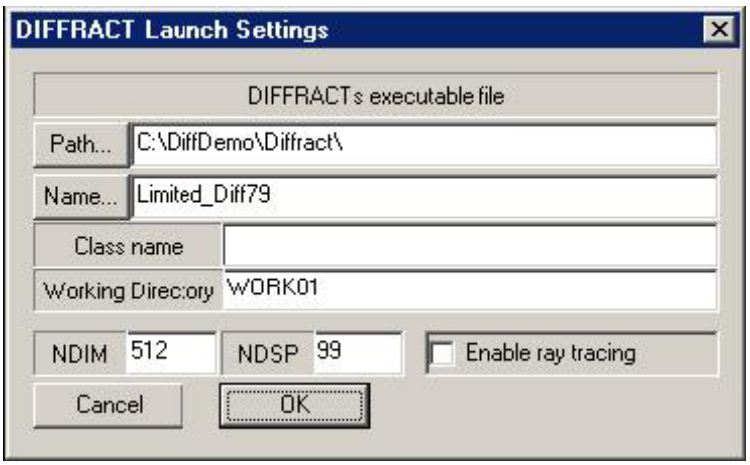


この会話ボックスを OK を押して閉じる。すると他のウインドウ **Workbench** が開き、シミュレートするシステムの図を表示する。（この図は **Workbench** と呼ばれる。）



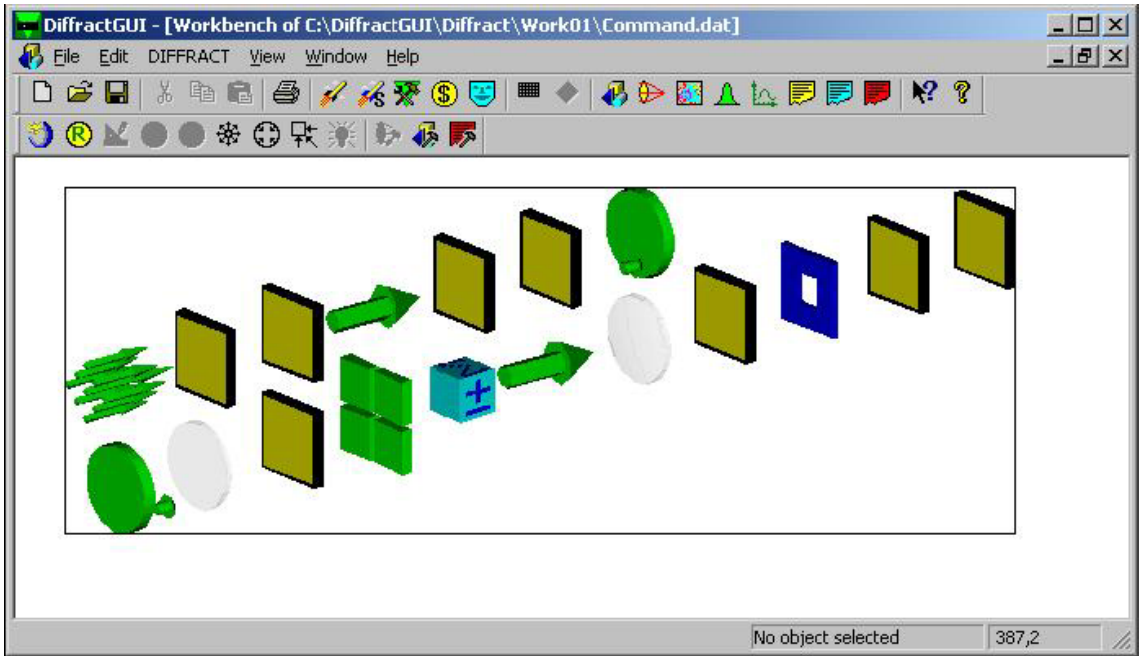
3. 光学系の Command.dat ファイルが開かれると、Workbench ウインドウは対応する Workbench を表示する。Menu バーの DIFFRACT を指定し、Settings を選ぶ。





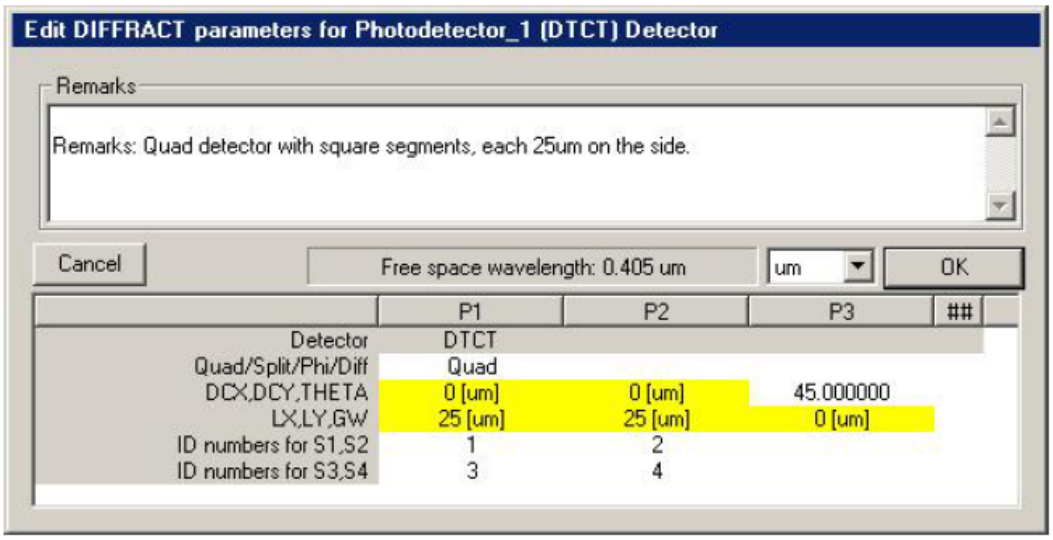
会話ボックスが開かれ、DIFFRACT で実行なファイル名、作業サブディレクトリ、DIFFRACT が実行されるとき使われるいくつかのパラメーターが表示される。(会話ボックスに現れるパラメーターの NDIM と NDSP はそれぞれシミュレーションで使われるメッシュの大きさと DIFFRACT で使われるいくつかの描画に使われるメッシュの大きさを表している。) これらのパラメーターは DiffractGUI によって自動的に設定されるが、必要があれば Diffract_Launch_Settings 会話ボックスで、ユーザーが修正することもできる。

4. **Workbench** ウインドウにおけるシステムの表示をよくするために、システム図面を制限しているボックスの外側をクリックし、キーボードの→↑←↓を使って Workbench を動かすか、Ctrl キーを押しながら→↑←↓を使って perspective を変えると良い。同じような操作をシステムの個々のエレメントにマウス左ボタンをクリックし (すなわち object を選んで) 矢印キー (Ctrl キーを押したり押さなかったりしながら) を使って選んだ object を移動または回転することができる。



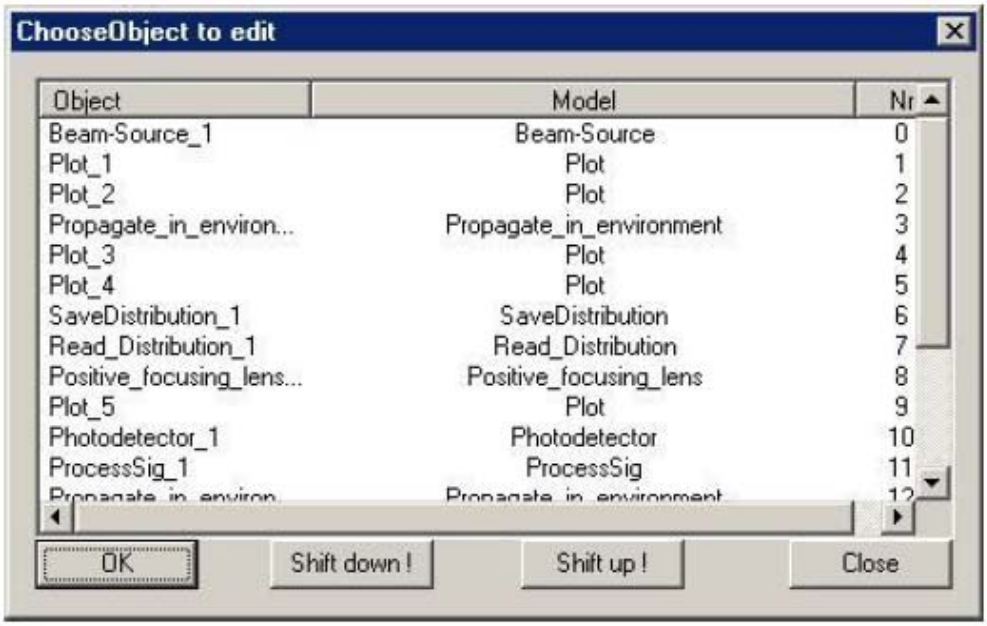
5. Workbench のそれぞれの object のパラメーターをチェックするためには、object をダブルクリックすればよい。Edit_DIFFRACT_Parameters 対話ボックスが開き、object に選んだ種々のパラメーターを見ることができ、これらのパラメーターをマウスの左ボタンでクリックし、希望の新しい値を選択または入力すればよい。DIFFRACT の言語では、*Edit_DIFFRACT_Parameters* 対話ボックスは、選択した object、たとえば BEAM, LENS, MASK, PLOT その他 DIFFRACT の menu で選択できる object に対応する command block を示す。シミュレーションにおける自由空間の波長 λ は、*Edit_DIFFRACT_Parameters* 対話ボックスの command block の最初に指示されていることに注意されたい。(たとえば mm から μm に) λ の単位を変更すると、対話ボックス内のすべての長さのパラメーターが自動的に変更されることを注意されたい。この特性は command block のはじめに長さの単位を決

定し、その後すべての長さのパラメーターを任意の単位（たとえば、cm, mm, μm, nm, あるいはλ、すなわち光源の真空中の波長で割り算した基準化された単位）で入力することができる。



6.

6. Workbench ウインドウ内で Workbench の外（あるいは Workbench 内であってもその中の object から離れたところ）をダブルクリックすると、*Choose_Object_to_edit* 対話ボックスを開くことができる。ここで再びある object を選びそのパラメーターを編集し、あるいは対話ボックスの下部にあるボタンを使ってその object の Shift Up または Shift Down を行うことができる。

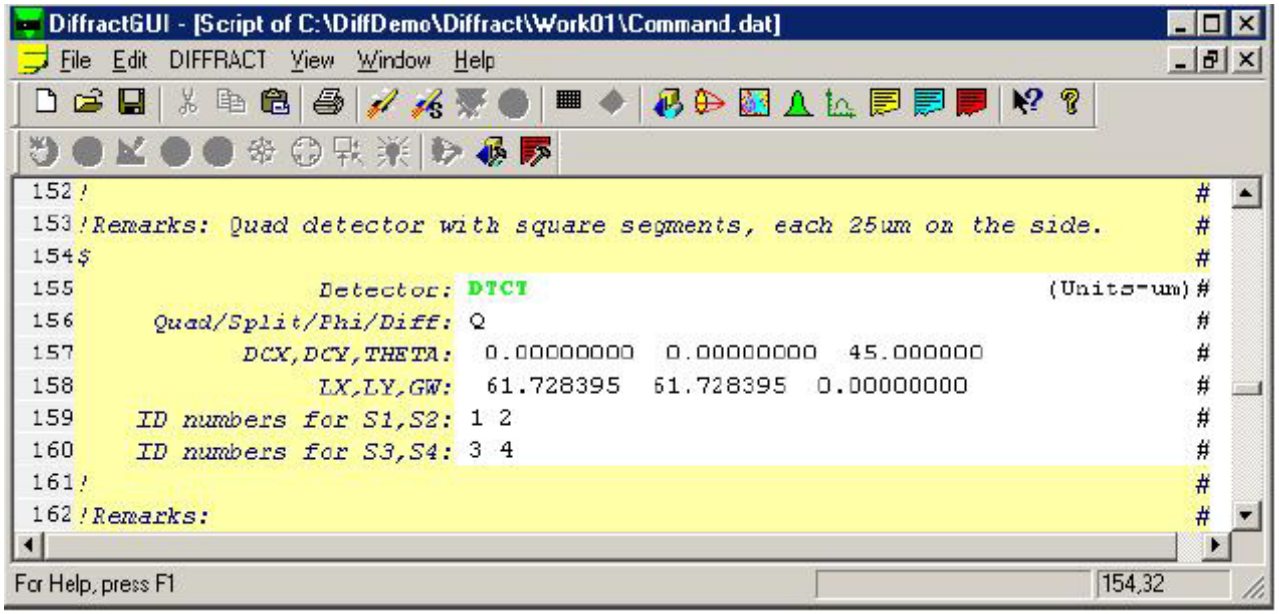


新しい object をシミュレーションを行うシステム（すなわち Workbench）に追加すること

ができる。それには Workbench でどれかの Object を選び、その後 menu バーにある New Object ボタンを押す。すると挿入可能の object のリストが現れるので、そのアイコンの一つをクリックして選択すれば、その object が先ほど選択した Workbench の object の直後に挿入される。

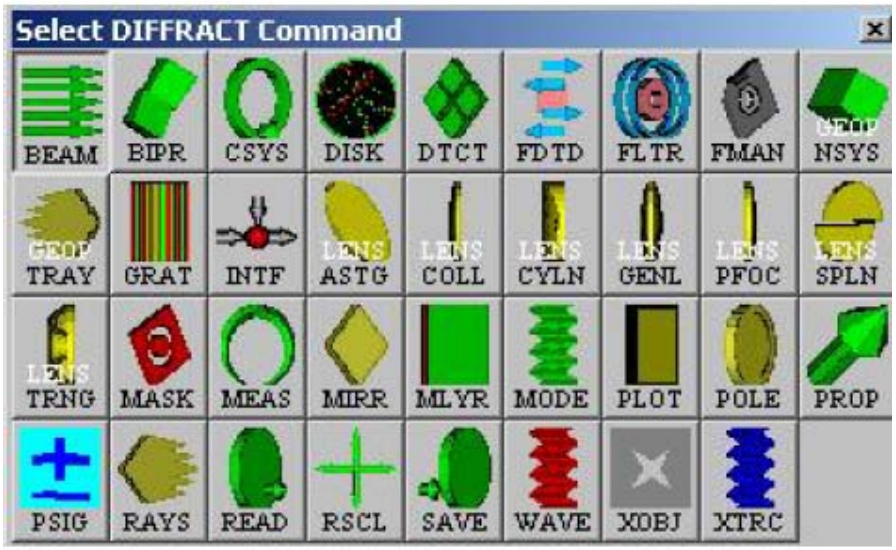
7. Munu バーのボタン *Inflare*, *Deflare*, および *Align_to_Top* は **Workbench** の大きさを変えたり、それを上左の隅まで動かすことができる。またカーソルでボックスをつかんで Workbench ウインドウを越えて動かすことができる。どれかの Object を動かしたり回転したりしても、Menu バーの Reposition ボタンを押せば、動かした object を元の位置・方向に戻すことができる。

8. もし Workbench のシステムに変化を加えたときは、それを **Command.dat** ファイルに保存するために Menu バーの Save ボタンを使わなければならない。Menu バーの View_Command.dat ボタンをクリックして、Command.dat ファイルの全体を見ていただきたい。Command.dat を開くと、それぞれの command block に対する長さのパラメーターがその Command block の上右の隅に(unit = xx)で表示された長さの単位で表されているのを見ることができる。(*Edit_DIFFRACT_Parameter* 対話ボックスで好きな単位に変更することができる。そうすると、次に *View_Command.dat* をクリックすると、対応する command block の最上段に新しい単位が表示されているのを見ることができる。)


DiffractGUI - [Script of C:\DiffDemo\Diffract\Work01\Command.dat]
File Edit DIFFRACT View Window Help
152 !
153 !Remarks: Quad detector with square segments, each 25um on the side.
154 $
155 Detector: DTCT (Units=um)
156 Quad/Split/Phi/Diff: Q
157 DCX,DCY,THETA: 0.00000000 0.00000000 45.000000
158 LX,LY,GW: 61.728395 61.728395 0.00000000
159 ID numbers for S1,S2: 1 2
160 ID numbers for S3,S4: 3 4
161 !
162 !Remarks:
For Help, press F1
154,32

9. **Workbench** に新しい object を挿入するには、まず Workbench から一つの object を選

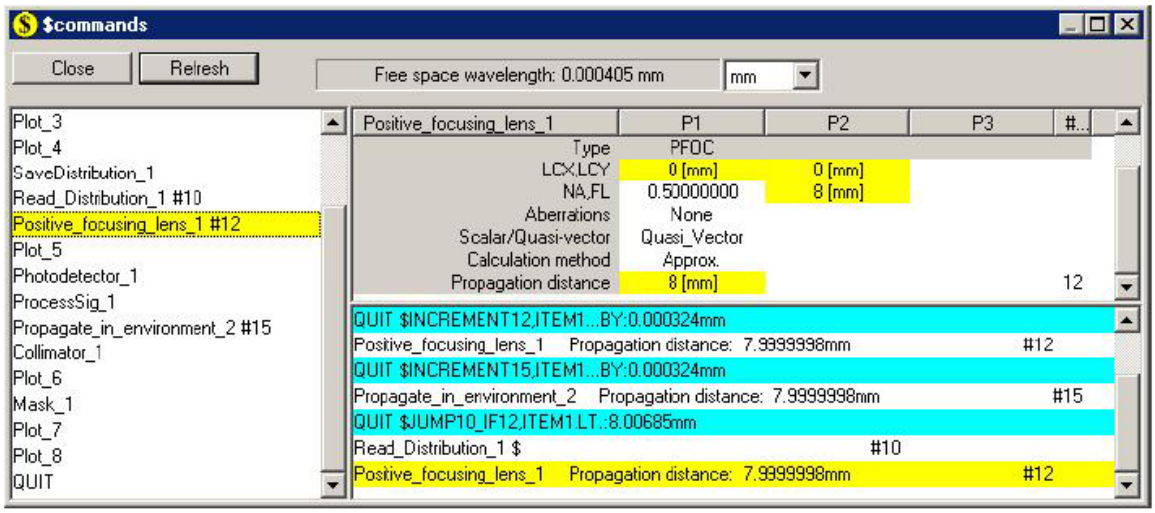
択し、menu バーの *Cmmand_Panel* ボタンをクリックする。挿入可能なすべての object のリストが表示されるので、挿入する object を *Command_Panel* から選べば、最初 Workbench で選択した object の直後に挿入される。



これがシミュレートするシステムに新しい object を挿入する最も簡単な方法である。事実、新しくファイルを開いて（Menu バーの *File_New* をクリックする）、希望の object を一つずつ *Command_Panel* から新しいファイルに入れる。その後、**Workbench** でその新しい object をダブルクリックし、そのパラメーターを *Diffract_Parameters* 対話ボックスからその command block に入れて、システムを構築することもできる。

10. *Workbench* 環境では、**Command.dat** ファイル内のあちこちに挿入した**$Command** を探すことができる。これらの$command は Command.dat ファイルにおけるループ演算を行うことができる：これらは種々の command block 内で数行の演算を参照している。行番号は**ユーザーによってそれぞれの行に割り当てられ**、各行の最後#記号の後に書かれている。Command.dat ファイルの右のマージンで＃記号の後に番号のついている行を探そう。与えられた行の特別なアイテムに標識をつける（**Tagging**）か、または与えられた行の特別なアイテムの値を増す**(Increment)**ことをプログラムに要求するか、あるいは Command.dat ファイルのある行へジャンプ（**Jump**）することをプログラムに要求するなど、$command は **Command.dat** ファイル内でループ演算を作るために強力なツールとなっている。Menu バーの*$Command_editor* は Command.dat ファイルの種々の場所に $Command を挿入することを補助する。ユーザーがしなければならないことのすべては、

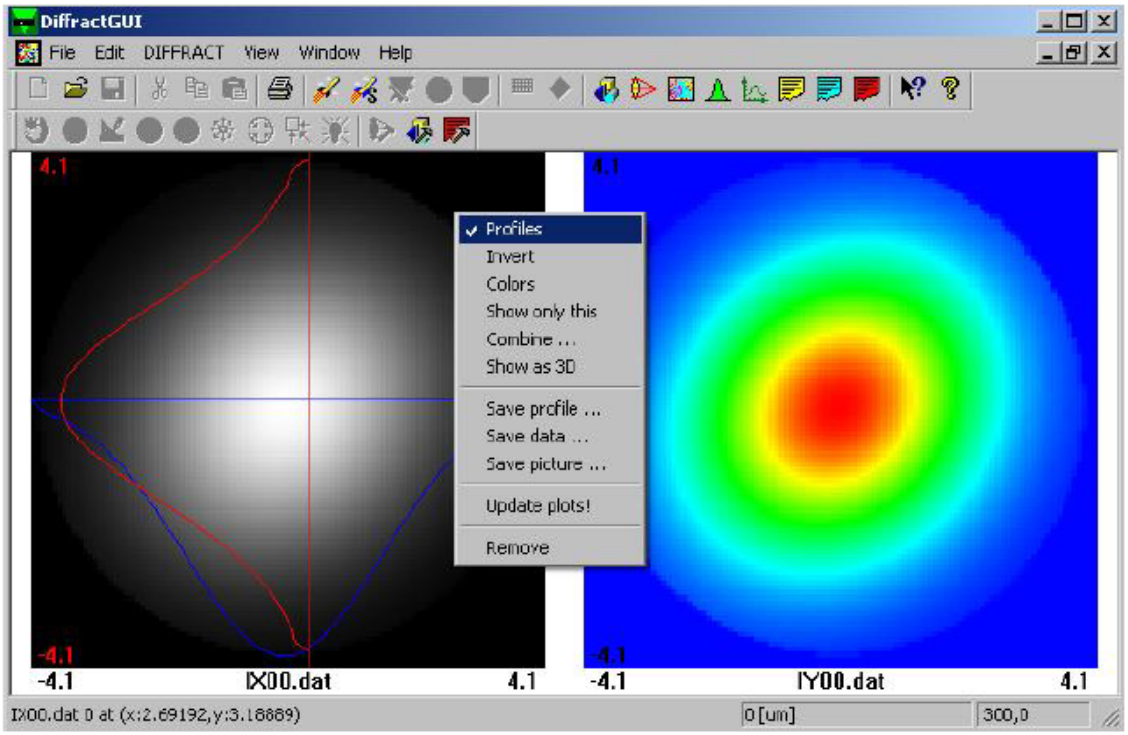
その直前に$command を挿入すべき object を選び、$command の最初の数文字を希望する場所に入力し、Enter を押せばよい。DiffractGUI の *$Command_editor* は正しい **$command** を特定の場所に挿入し、ユーザーにこの$command の未知のパラメーター（place-holder ? で印をつけられている）を特定の値に置き換えるように促す。

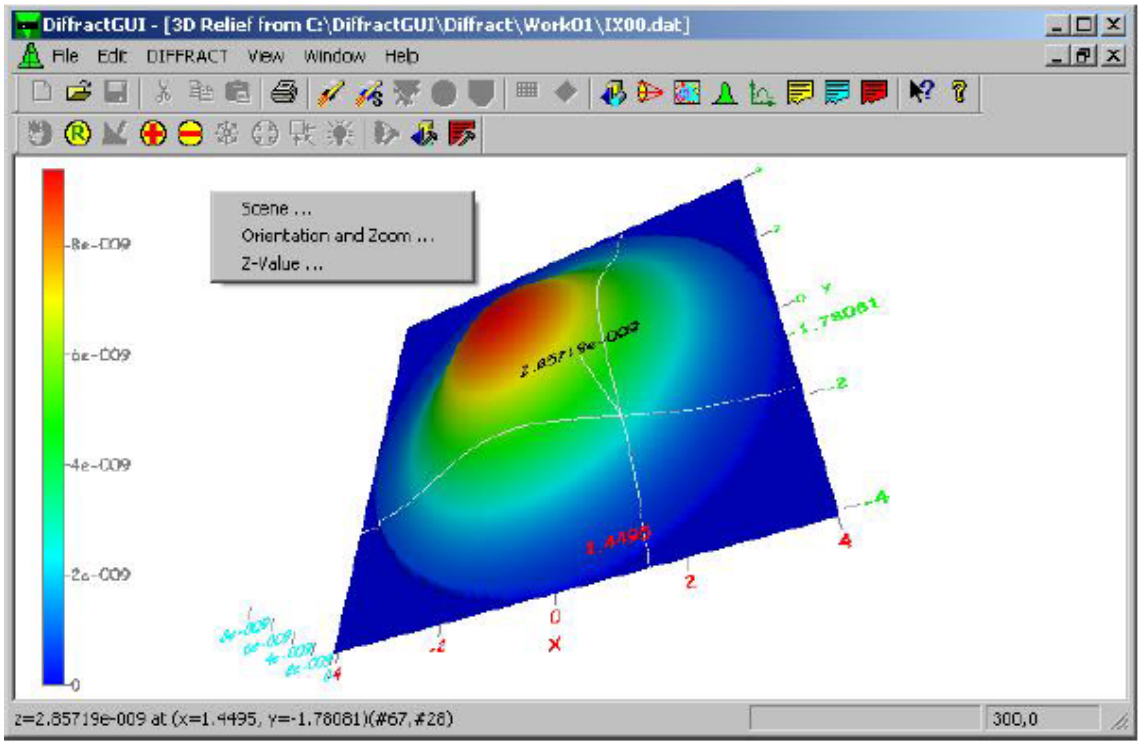


11. 一旦 **Workbench** にシステムが構成され、そのパラメーターが決まると、ユーザーは Command.dat ファイルを（Save ボタンを使って）保存し、*Launch_Diffract (Zoom)* または *Launch_Diffract(Step)* ボタンを使って DIFFRACT を実行する。Zoom モードでは、DIFFRACT は全 Command.dat ファイルを実行し、シミュレーションが完了してから停止する。**STEP** モードでは、Command.dat ファイルを step-by-step に実行し、プログラムの入力・出力パラメーターを一度に 1 ページとして表示し、ユーザーが Enter キーを押して始めて 1 段の演算が行われる。STEP モードで実行すれば、ユーザーがシミュレーションをチェックし、各ステップが正しく実行されたかを確かめることができる。ZOOM モードで実行すれば、シミュレーションを全体としてチェックし、ユーザーが干渉することなく実行できる。特に、ループ演算が Command.dat ファイルに組み込まれた場合、ユーザーは演算を *Launch_Diffract(Zoom)*に任せ、シミュレーションの間席を離れることができる。

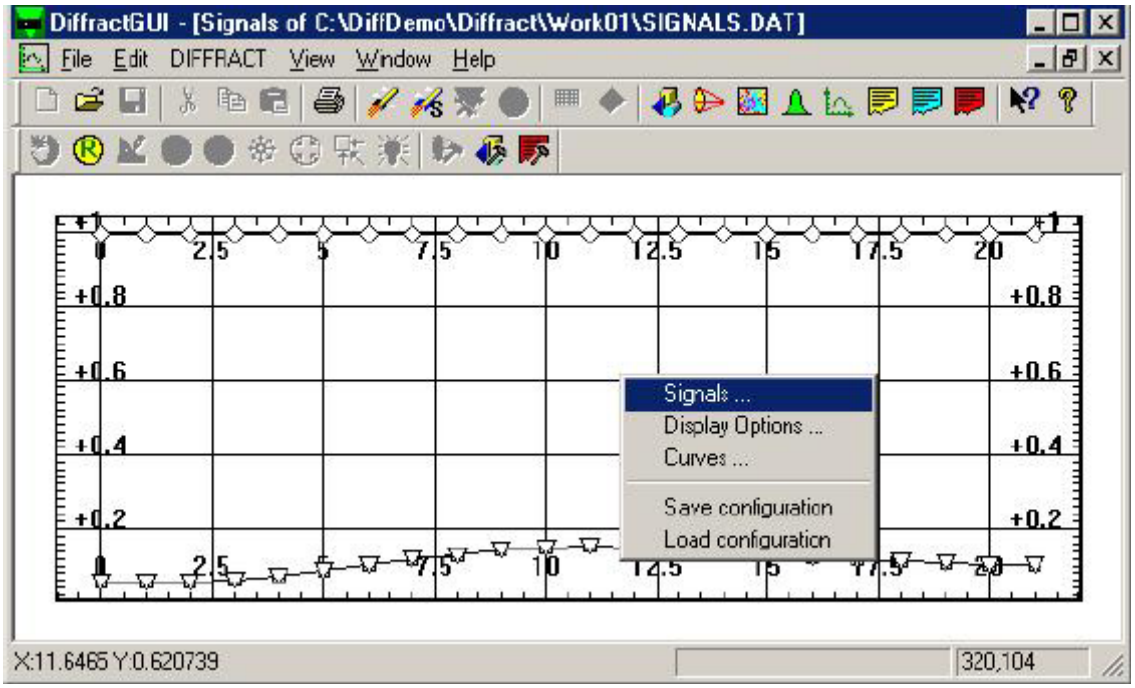
12. シミュレーションが終わると、すべてのグラフィックファイル、Session.dat ファイルおよび（もしできれば）Signal.dat ファイルは作業ディレクトリの中に保存される。この時点で、ユーザーは **Workbench** から **PLOT** object を選び、bitmap で描画するために *View_Bitmap* をクリックする。（**PLOT** object はシステムにおける種々の断面の強度分布、

対数強度分布、位相分布、偏光分布、あるいは幾何光学光線分布を表示するために用いられる。）これらの bitmap 像でマウスの右ボタンをクリックすると、可能な表示の option が示される。このリストを使うと、ユーザーは bitmap 像をカラーまたはグレイスケールで表示することができ、また 3 次元表示、反転カラー表示などができる。同様にユーザーは Workbench で **PLOT** object を選び、*View_Relief* ボタンを押せば対応する分布の 3 次元表示を見ることができる。





もし **Signal.dat** ファイルがシミュレーションの過程で作られたのであればーこれは一般に **Process_Signal** object が **Detector** 信号と Siganl.dat ファイルへの出力をプリントする場合であるがーmenu バーの *View_Signal* ボタンを押せば計算された信号を描画することができる。信号ウインドウでマウスを右クリックすれば、描画処理の option を選ぶことができる。



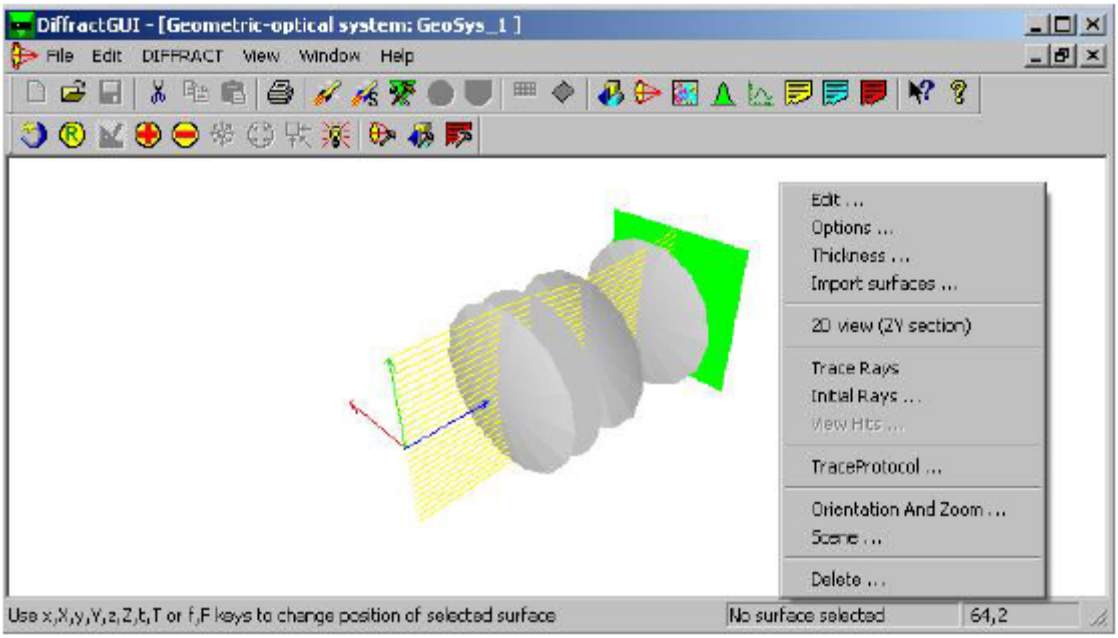
シミュレーションを行えば、作業ディレクトリ内に Session.dat というテキストファイルができる。Menu バーの *View_Session.dat* ボタンを押せばこのファイルを表示することができ、ユーザーはシミュレーションにおけるすべての入力パラメーターの数値と、いくつかの主要な出力パラメーターを見ることができる。

13. Menu バーにおけるその他のボタン（*View_GEOP*, *Trace_rays*, *Scale_Up*, *Scale_Down*）を使うと、**Geometric-optical System**(GEOP)を操作することができ、その断面の 2 次元または 3 次元表示、システムを通過する多数の光線の明示とそれらの追跡、種々の面における光線の到着点の観察などを行うことができる。簡単な Geometric-optical_System の例を Test_run3(作業ディレクトリ C:¥DiffractGUI ¥Diffract¥Work03 内にある)に示す。

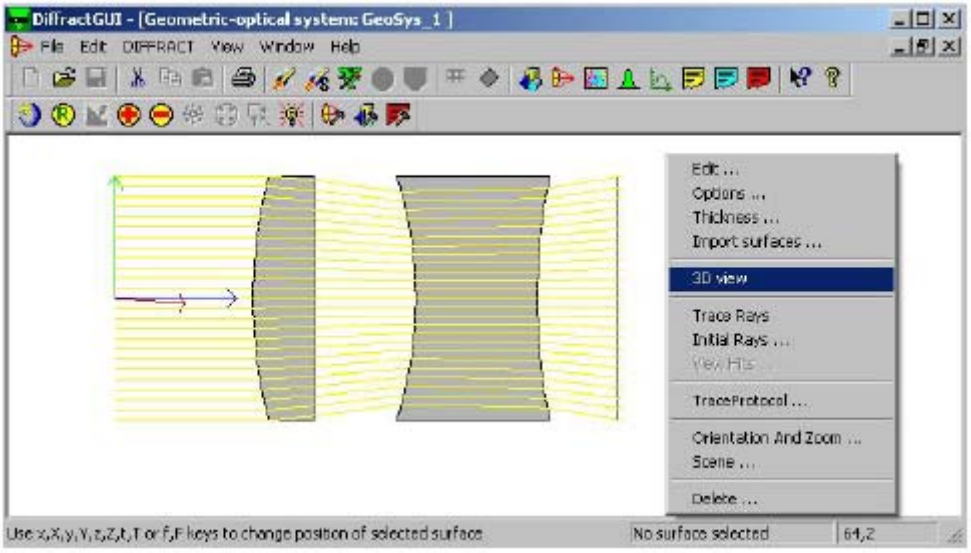
<Work03>内の Command.dat ファイルを開き **Workbench** 上の Geometric-optical_System (GEOP object)を観察する。GEOP object 上でマウスの左ボタンをクリックし、次いで **GeoSysView** を開くために **View GEOP** ボタンを押す。この図の中にこの GEOP システムの 4 個のガラス面が（相対的に）正しい位置と大きさ（GEOP object command block に規定されたような）で表示される。

ScaleUp および **ScaleDown** ボタン、あるいは簡単にはマウスホイールは、GeoSys View でシステムの拡大・縮小に使うことができる。またマウスカーソル（Ctrl キーを押したり押さなかったりしながら）を使ってシステムを元の位置・方向に戻すのに用いられる。すなわちウインドウの背景でマウスの左ボタンを押したままで、マウスカーソルを動かす。

必要な場合には Edit バーの **Reposition** ボタンをシステムの再配置に使うことができる。



GEOP 光学面の一つをダブルクリックし、その光学面の **Edit DIFFRACT** Parametrs 対話ボックスを開くと、必要があればそのパラメーター値を変更することができる。ウインドウの背景をダブルクリックすると、GEOP 全体の **ChooseObject to Edit** 対話ボックスを開くことができる。ここで再び menu からどれかの光学面を選ぶと、そのパラメーターを変えることができる。**GeoSysView** にあるときは、ウインドウのどこでも右マウスボタンをクリックすると、**pop-up menu** が開き **2D VIEW** を選ぶと GEOP システムの YZ 面の断面を観察することができる。再び pop-up menu を使い、今度は **3DView** を選ぶとシステムの 3 次元描画を得ることができる。



（右マウスクリックで開く）pop-up menu は、いくつかの option を可能とする。その中に **Initial Rays** がある。この option を選ぶと対話ボックスが開く。dx, dy, dz, $d\theta$, $d\phi$ に適当な値（たとえば 0.2, 0.2, 0, 0, 0）を選び、**Add** ボタンを数回押すとシステムを追跡する何本かの initial ray を作ることができる。再び右マウスボタンをクリックし、今度は **Trace Rays** option を選ぶ。（同じことが Edit バーの **TraceRays** ボタンを押しても実行できる。）今や GEOP 光学面のどれでも選んで（右マウスボタンをクリックして）pop-up menu を開き、その面の **View Hits** を見ることができる。[1]

[1] DIFFRACT の Limited Edition は **RAYS** option を実行できない。この option は通常 **Geometric-optical System** への入射ビームから一組の光線を作る。そのため、Limited Edition は GEOP における波面の追跡ができない。そこでユーザーは DiffractGUI の中で分離した **GEOP** を作り、それを 3 次元あるいは断面（2 次元）で観察し、数本の光線を選んで光学系を通過させて解析する。しかし、そのような GEOP は GEOP 全体を波面で追跡することが必要な完全なシミュレーションはできない。これは DIFFRACT の *General Edition* と *Student Edition* の特性である。

2. DiffractGUI のインストール（Installing DiffractGUI）

DiffractGUI をインストールするには、配布された CD-ROM に切り替え、DiffractGUI のアイコンをクリックされたい。このディレクトリの中に入ったら、Setup.exe をダブルクリックし、あとはモニター上の指示に従うとよい。ハードディスクには DiffractGUI のためのホームディレクトリ、たとえば C:¥DiffractGUI を作ると、セットアップによってすべての必要なファイルとフォルダーをこのディレクトリ内に転送され、Windows 環境下で実行可能の準備が行われる。このフォルダーには下記のようなファイルとサブフォルダーが含まれている。

- **<Diffract>** Diffract.exe, Settings.log, その他いくつかの作業サブディレクトリ。
- **<GeoModels77>** 同様な名前のファイル 4 個。
- **<GoeModels79>** 同様な名前のファイル 4 個。
- **<GeoModels81>** 同様な名前のファイル 4 個。
- **<Manuals>** QuickGuide.doc および UserGuide.doc。
- **<Models77>** Mnn_xxxx.m3d という名前のファイル 31 個。
- **<Models79>** Mnn_xxxx.m3d という名前のファイル 35 個。
- **<Models81>** Mnn_xxxx.m3d という名前のファイル 35 個。
- **<Models82>** ABCD_xxxx.m3d という名前のファイル 35 個。
- **<Models83>** ABCD_xxxx.m3d という名前のファイル 35 個。
- **<Rainbow>** Hard_key のドライバーSetup.exe と Readme.doc。
- **<TestRuns77>** デモファイルとして、いくつかの command.dat ファイル。
- **<TestRun79>** デモファイルとして、いくつかの command.dat ファイル。
- **<TestSystems>** デモファイルとして、いくつかの geometric-optical system。
- **DiffractGUI.cnt** ヘルプファイルの目次。
- **DiffractGUI.exe** ただ 1 個のプログラム実行ファイル。
- **DiffractGUI.hlp** ヘルプファイル。
- **Logfile.txt** セッションのログファイル。
- **ReadMe.txt** ユーザー向けの最新情報。
- **Uninst.isu** DiffractGUI を削除するために必要な情報。

(DIFFRACT の Graphic User Interface)**DiffractGUI.exe** はそのアイコンをダブルクリックすれば起動する。

<Diffract>には、DIFFRACT の実行ファイル DIFFRACT.EXE, SETTINGS.LOG, および 3 個のサブディレクトリ<WORK01>,<WORK02>, <WORK03>が含まれている。この 3 個

のディレクトリはそれぞれ **COMMAND.DAT** ファイルを持ち、これは DiffractGUI の学習用に使える。ユーザーが演算を行うときは、<Diffract>の中に作業ディレクトリを作っていただきたい。たとえば、C:¥DiffractGUI¥Diffract¥JohnDoe というディレクトリを作り、このフォルダーの中にシミュレーションの COMMAND.DAT をおいて頂きたい。

<Manuals>フォルダーは、Graphic User Interface のための **User's Guide** を含んでいる。"Instruction to set up and run simple simulation within DIFFRACT's Graphic User Interface(GUI)と題する第 1 章は DiffractGUI を知るためのよい出発点となるが、詳しいことはもうひとつのマニュアル UserGuide.doc の対応部分を参照していただきたい。<Modelsnn>と<GeoModelsnn>は DiffractGUI を作動させるのに必要なファイルであるため、その内容やそれに含まれるファイルの内容を変えてはいけない。<TestRuns>と<TestSystems>は学習用の資料である。

3. 序論（Introduction）

DiffractGUI は DIFFRACT を容易に、また使いやすくするために開発された。このプログラムは主として DIFFRACT の command.dat ファイルを取り扱う。*DiffractGUI* は COMMAND.DAT ファイルの内容を表示したり、種々の方法で変更したりすることができる。さらに、幾何光学システムを 3 次元的に表示したり、変更したり、また DIFFRACT の出力ファイルを bitmap や 3 次元的に表示できる。DIFFRACT のセッションの間にできるすべてのファイルは、DiffractGUI 内部で見ることが出来る。

プログラムは MDI-Application としてデザインされ、Windows 98, NT, Me, 2000, XP で実行できる。このことは、同時に DiffractGUI 内部で開く一つ以上の COMMAND.DAT ファイル（これ以後は“document”と書くことにする）ができることを意味する。そのような document のそれぞれに対し、8 個のビューウインドウが用意され、それぞれ独立に現れたり、隠したりすることができる。これらのビューの幾つかは、COMMAND.DAT の内容を（異なったモードで）表示することができるが、他のビューは DIFFRACT の出力ファイルを開いて表示するのに使われる。

8 個のビューウインドウは LoggerView, View3D, ScriptView, GeoSysView, SignalView, ReliefView, SessionView および BitmapViwe であって、これらの詳細は後で述べる予定である。

3.1 モデルファイル（Model Files）

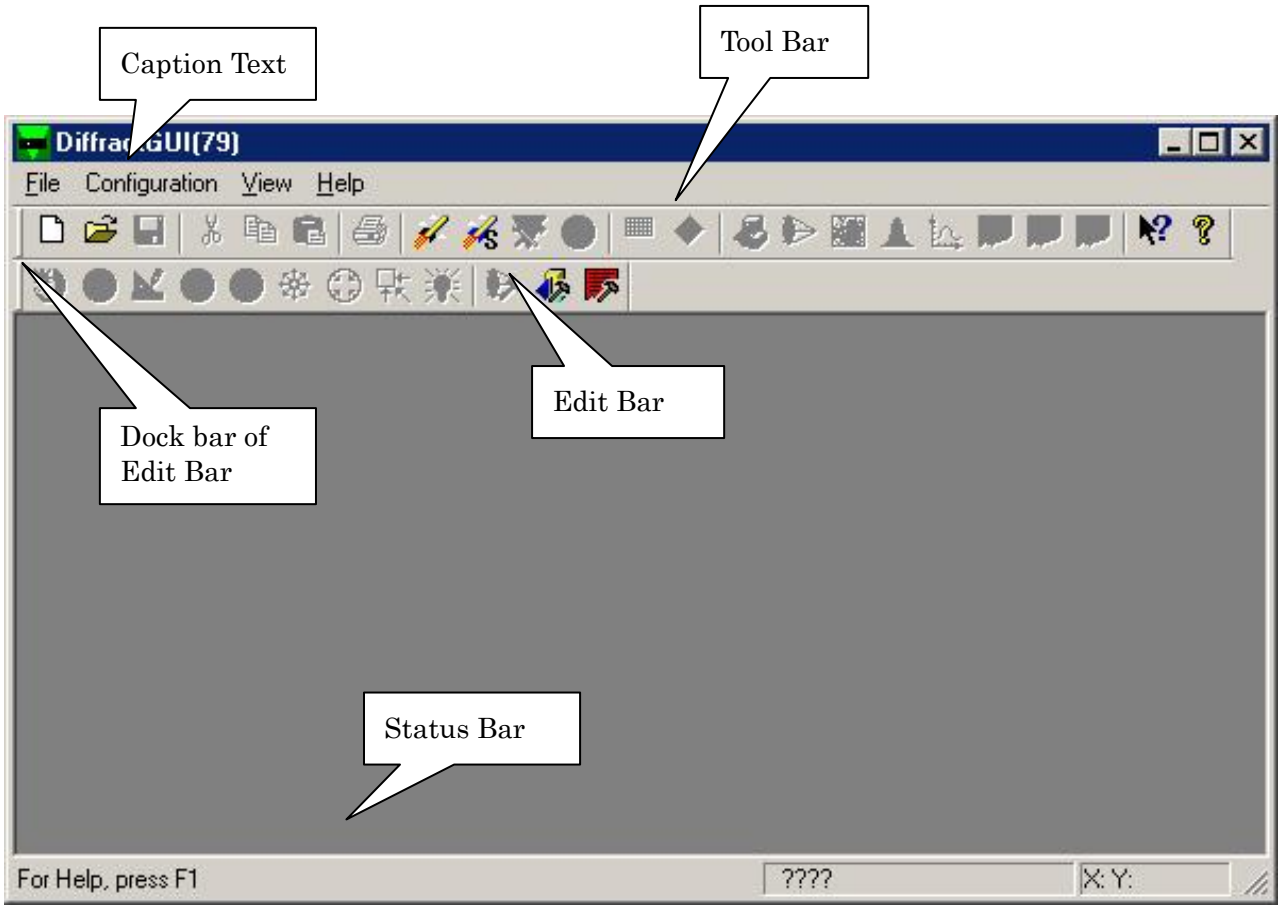
COMMAND.DAT ファイルを読み込むと、*DiffractGUI* は内部でファイルを 1 つまたはそれ以上の文書のブロックに分割する。そのようなブロックのそれぞれは、DIFFRACT の menu 命令（たとえば、**BEAM, LENS, PLOT,** など）とそれらに関連するパラメーター値のリストをもち、DIFFRACT のパラメーターとして参照する。*DiffractGUI* のインタープリターはどのような menu 命令が定義されているか、またそれぞれの命令に対し DIFFRACT パラメーターの順序、大きさ、およびタイプを知らなければならない。DIFFRACT の将来の Version up に際し、DiffractGUI が柔軟に対応するために、この情報は外部のいわゆる model ファイルに収めてある。それぞれの DIFFRACT の Version に対して、1 個の model ファイルがなければならない。すべての model ファイルは **m3d** という拡張子を持ち、単一のフォルダー内に置かなければならない。このファイルの名前も位置も任意であるが、ユーザーはこれを View3D Option を用いて DiffractGUI 内に設定しなければならない。デフォルトではこのフォルダーは **C:¥Diffract¥Modelsnn** となっていて、**nn** は DIFFRACT の Version（DIFFRACT 7.7, 7.9, 8.1, 8.2 または 8.3）から取っている。

新しい documet を作る時は、DiffractGUI は model フォルダーの中のすべてのファイルをスキャンし、それによって COMMAND.DAT のインタープリーターを update する。

同じように、光学面に対する model ファイルは geometric-optical システムのファイルの中になければならない。（現在、そのような光学面は 4 個ある。**Aperture**(開口), **Glass**（ガラス面）, **Mirror**（鏡）および **Destination plane**（最終面）である。）幾何光学システムファイルを開いて表示した時は、そのフォルダーは DiffractGUI の GeoSys インタープリーターが必要とする情報を読み込まなければならない。このフォルダーは、前に述べた model フォルダーと同じように、デフォルトでは **C:¥Diffract¥GeoModelsnn** となっている。もし、このフォルダーの名前と位置が変更されるときは、ユーザーは GeoSysView Option を用いてこのことを DiffractGUI に知らせなければならない。

3.2 プログラムの起動（Start Up）

DiffractGUI をスタートすると、小さいタイトル・ウインドウが数秒間スクリーンに現れ、その後 main window が開く。タイトル・ウインドウはその上をマウスでクリックするか、return キーを押すか、あるいは数秒間待てば消失する。次の図は start-up View(document は何も開いていない)と、Diffract の主なビジュアルエレメントが見えている。
この環境で動作させるには、ユーザーは既に存在している document ファイルを開く(menu bar から **File-Open** を使う)か、あるいは新しい empty file を作る（**File-New** を使う)。Document の作成または修正を行った場合には、**File-Save** または **File-SaveAs** を使って保存しなければならない。
Tool bar, Edit bar および Status bar は **View-ToolBar**, **View-EditBar** および **View-Status Bar** によって使用/不可能にすることができる。Tool Bar と Edit Bar は元の位置から動かすことができ、マウスの左ボタンを押しながらドラッグして、Window の任意の辺側に置いたり、floating window として固定せずに適当な位置に置くことも出来る。



3.3 Edit Bar（編集バー）

Edit Bar は Edit の menu のうちよく使う命令をショートカットボタンとして配置している。

Icon	Menu 命令	内容
	Edit Create	新しい object や光学面を作って挿入する
	Edit Reposition	スクリーン表示をやり直す
	Edit Scaleup	スケールファクターを増大する
	Edit ScaleDown	スケールファクターを減少する
	Edit Inflate	object または面のディスプレイサイズを大きくする
	Edit Deflate	object または面のディスプレイサイズを小さくする
	Edit Align To Top	図面をスクリーンの左上に置く
	Trace Rays	GeoSysView で光線追跡を行う
	View-Setting-GeoSys	GeoSysView の起動
	View-Setting-Workbench	Workbench の起動
	View-Setting-Logger	LoggerView の起動

3.4 Tool Bar(ツールバー)

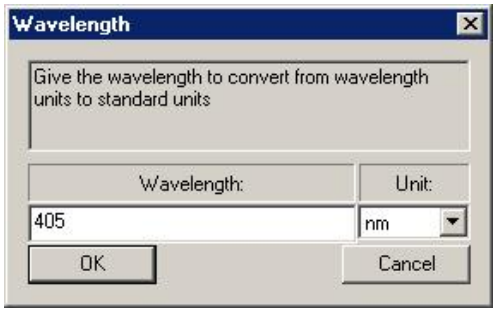
Tool Bar には、DiffractGUI menu において頻繁に使う命令をショートカットボタンとして収容している。

Icon	Menu command	内　　容
	File New	新しい空の document を作る。
	File Open	ディスクからファイルを開く
	File Save	現在開いているファイルを保存する
	Edit Cut	選んだ対象を View からカットする
	Edit Copy	選んだ対象を clip board にコピーする
	Edit Paste	Clip board の中味を View へ貼り付ける
	File Print	現在の View を印刷する
	DIFFRACT Launch (ZOOM)	DIFFRACT をズームモードで実行する
	DIFFRACT Launch (STEP)	DIFFRACT を STEP モードで実行する。
	DIFFRACT Edit Parameters	**Edit DIFFRACT parameter** dialog を実行する
	DIFFRACT-$-command	$命令を実行する
	View-Command panel	Command panel を開く
	View-Surface panel	Surface panel を開く
	View-Workbench	**Workbench** viewer を開く
	View-GeoSys	**GeoSysView** を開く
	View-Bitmap	**BitmapView** を開く
	View-Relief	**ReliefView** を開く
	View-Signals	**Signal-View** を開く
	View-Script	付属の **COMMAND.DAT** を開く
	View-Errorlogger	**LogView** を開く
	View-Session	**SESSION.DAT** ファイルを開く
		環境に沿ったヘルプ
	Help-About	Version と serial number を表示する

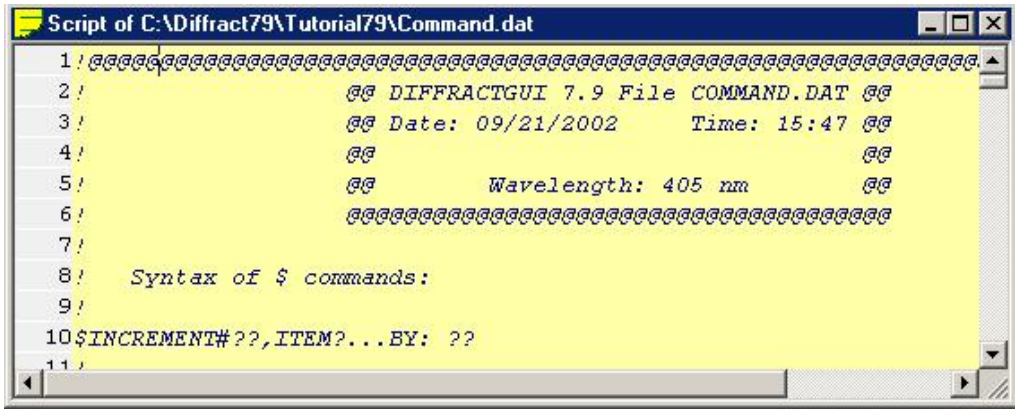
3.5　長さの単位の取り扱い(Handling Length Units)

DIFFRACT の早い Version(7.7, 7.9, 8.1)では計算に使用するすべての長さは波長で基準化されている。実際には、この制約は、他の単位で表されたシステムのシミュレーションを行う時に、多量の計算が必要となる。この複雑さを避けるために、*DiffractGUI* はすべての長さを mm，μm あるいは nm で扱う可能性を提供している。

新しい COMMAND.DAT ファイルが開かれたり、作られたりする時はいつでも、ユーザーは波長を mm，μm あるいは nm で表すように求められる。これは長さのパラメーターを基準化するのに用いられる。COMMAND.DAT ファイルを保存する時、すべての長さは基準化され、波長単位で保存される。波長が変化しない限り、COMMAND.DAT ファイルはセッションの間中、このことで影響が出ることはありえない。



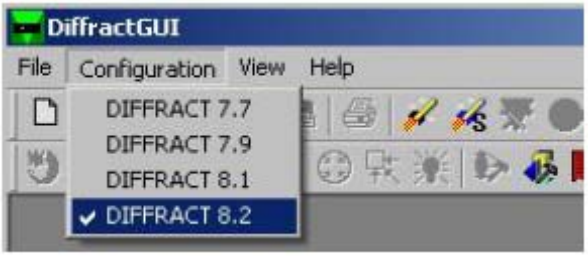
DiffractGUI は特定した波長を COMMAND.DAT ファイルの Header 部に保存する（header 部は DIFFRACT では無視される）。この COMAND.DAT ファイルを再度読み込むと波長の値を Header から読み取り、デフォルト値として取り扱う。


Script of C:\Diffract79\Tutorial79\Command.dat
1 !@@@

option は与えられた寸法の固定されたシステムを異なった波長で調べたい時、有用である。波長だけでなく、メトリック単位での長さのパラメーターの変更が可能である DIFFRACT 8.2 では、上記の形式はもはや使われていない。しかし DiffractGUI では前の Version と同様に使用できる。

3.6 プログラムの構成（Configurations）

DiffractGUI は Diffract 7.7, 7.9, 8.1, 8.2, および 8.3 の Version に従って、プログラムの構成を変えることができる。希望する構成は、menu **Configuraion** で決定することができる。この menu は *DiffractGUI* に何も読み込んでいない時に実行できる。



新しい構成を選択した時、*DiffractGUI* は次のようなメッセージを表示する。

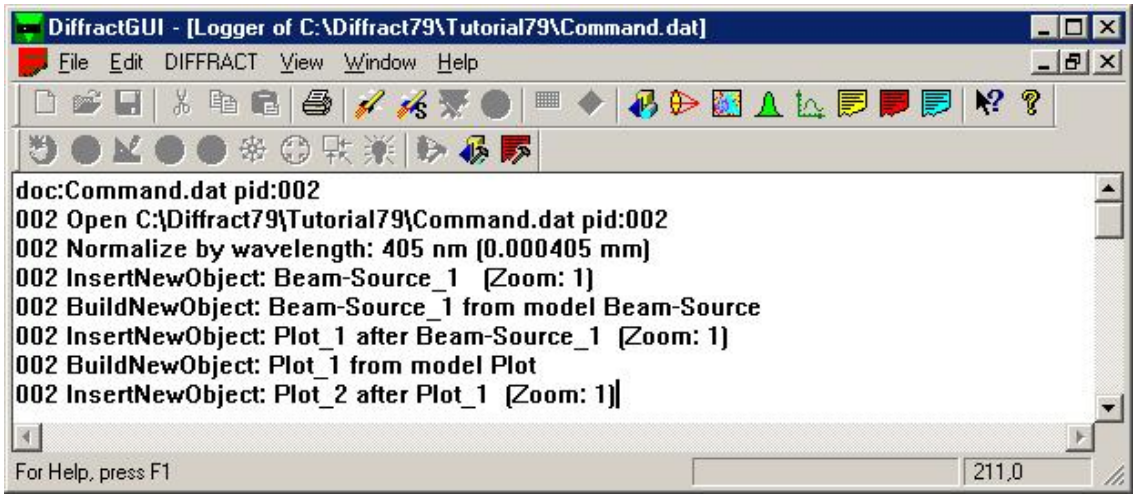
このメッセージボックスを"OK"で閉じると、*DiffractGUI* は停止するので、手動で再起動しなければならない。内部的に行ったすべての設定はそれぞれの構成に対して、別のレジストリーの登録として保存される。これは、もし構成を変えた場合は、（*DiffractGUI* を再起動したとき）model-path, DIFFRACT が実行できるファイルへの path などの内部設定を確かめなければならない。

3.7　フォント（Fonts）

Main menu command (**View-Setting-Font…**)から **ChooseFont** 対話ボックスを開き、フォントの変更ができる。選択されたフォントは Nenu command によって開かれた View 画面で使われる。Menu command が灰色（不可能）の場合には、フォントの変更はできない。フォントの変更が可能な View は、**ScriptView, SignalView, SessionView,** および **LogView** である。この特性は多くのフォントが表示されないその他の View (**Workbench, GeoSysView**, および **ReliefView)**には適用されない。(註 : **ScriptView** は正常に動作するためには"Fixed_pitch"フォントが必要である。したがって、**ScriptView** のフォントを変更しようとして **ChooseFont** 対話ボックスを開くと、適用可能のフォントリストには fixed_pitch フォントのみが載せられている。通常パーソナルコンピューターにはそのようなフォントは多くインストールされていないので、他の View の **ChooseFont** より短いリストが表示される。)

4.　Logfile の観察（The LoggerView）

DiffractGUI が起動すると、(background で）新しいテキストファイルを作り、エラーレポートやセッションの間に開かれたすべての documents に関する情報を記録する。*DiffractGUI* はセッション中に多くの documents を扱うが、しかし唯一つの logfile(記録ファイル)があり、LoggerView はその時点でアクティブな document に関連したこれらのメッセージのみを見ることが出来る。この記録ファイルは *DiffractGUI* の新しいセッションが始まると上書きされるので、ユーザーがこのファイルを残したい時は、*DiffractGUI* のセッションが終わった時点で、その名前を変えるか、他のディレクトリへ移す必要がある。

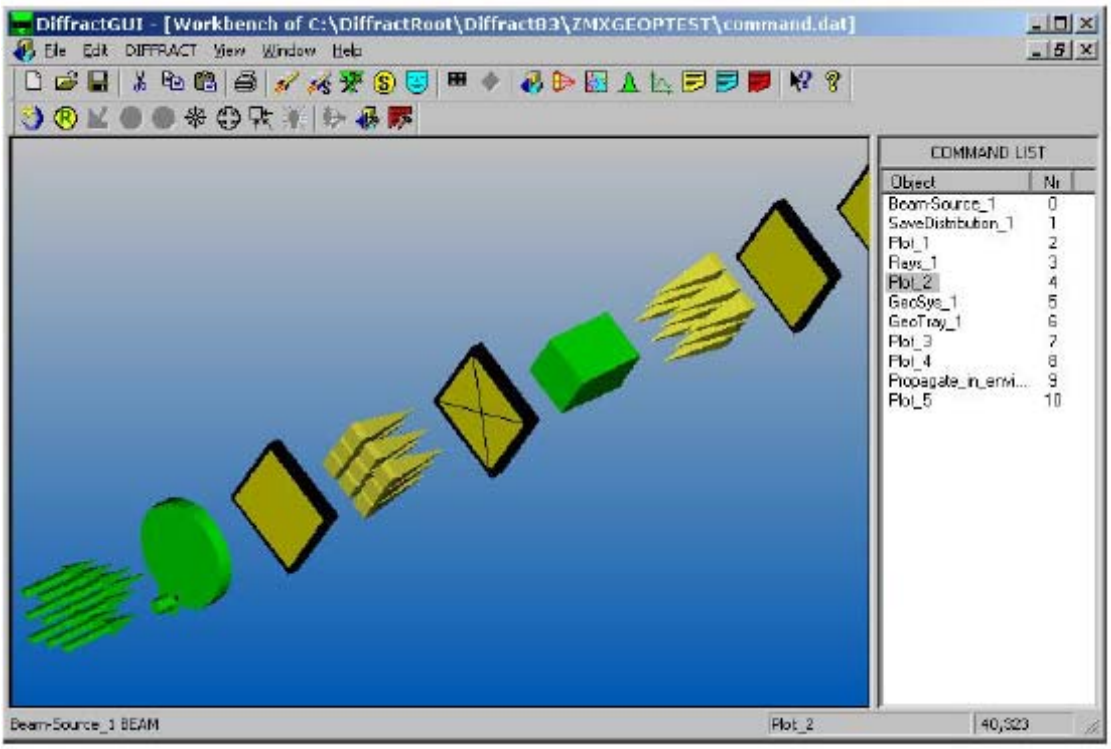

DiffractGUI - [Logger of C:\Diffract79\Tutorial79\Command.dat]
File Edit DIFFRACT View Window Help
doc:Command.dat pid:002
002 Open C:\Diffract79\Tutorial79\Command.dat pid:002
002 Normalize by wavelength: 405 nm (0.000405 mm)
002 InsertNewObject: Beam-Source_1 (Zoom: 1)
002 BuildNewObject: Beam-Source_1 from model Beam-Source
002 InsertNewObject: Plot_1 after Beam-Source_1 (Zoom: 1)
002 BuildNewObject: Plot_1 from model Plot
002 InsertNewObject: Plot_2 after Plot_1 (Zoom: 1)
For Help, press F1
211,0

4.1 Logger View の option (LoggerView Option)

View-Settings-Errorlogger という menu によって対話ボックスが開かれる。ユーザーは記録ファイルに新しい path と名前を書き込み、新しいファイルを決めることができる。対話ボックスが”OK”で閉じられると、新しい設定がレジストリーに記録される。*DiffractGUI* は、こうして作ったファイルを開いて記録を行うが、同じ名前のファイルが存在している時は、上書きされてしまう。もし、ファイルを開くことができないときは、*DiffractGUI* はエラーメッセージを出し、記録を止める。この場合、特定したフォルダーが存在するかどうか、またはそのファイルが“Read only”（読み取り専用）とマークされていないかを確かめる。

5. Workbench

この観察ウインドウで読み込まれた COMMAND.DAT に含まれている DIFFRACT の menu item を 3 次元的に表示できる。シミュレーションを行うシステムの object*(COMMAND.DAT に記録されたのと同じ順序で配列され、1 本の直線状の軸に沿って配置される。この軸の最初に置かれるのは BEAM または READ である。Viewer のインタープリーターが別の BEAM または READ を見つけると、新しい object を並べる軸を前述の軸の下に作る。それぞれの軸の方向は先頭に置かれた BEAM または READ の方向によって決定する。object を配置する間隔は一定（デフォルト値＝100pixels）である。



それぞれの object の特性は付属の model ファイルに記録されている。*DiffractGUI* はそれぞれの object に対して独特の名前をつけていて、それはマウスを object の上に置けば StatusBar の左側に表示される。そこでマウスをクリックすればその object が選択される。object の名前が StatusBar に表示され、別の object を選択するまでその状態が継続される。

5.1 object のチェック （Control Elements）

Workbench の内部で、object のアイコンの上でマウスを**左クリック**すればその object が選

*光学系を構成するレンズや鏡などの素子を指す。この Users Guide ではこれらの素子類を”object”という言葉で表している。

択される。**左ダブルクリック**によって Edit-DIFFRACT_parameters 対話ボックスが開かれ、この object に関するパラメーターが表示され、必要があればこれを変更することができる。これがユーザーが COMMAND.DAT ファイルの修正を行う唯一の方法である。
右ダブルクリックを行うと ScriptView が立ち上がり、この object に対する DIFFRACT の menu command を選ぶことが出来る。この object の選択は、**TAB** または **Shift+TAB** キーによって次の object に移ることもできる。

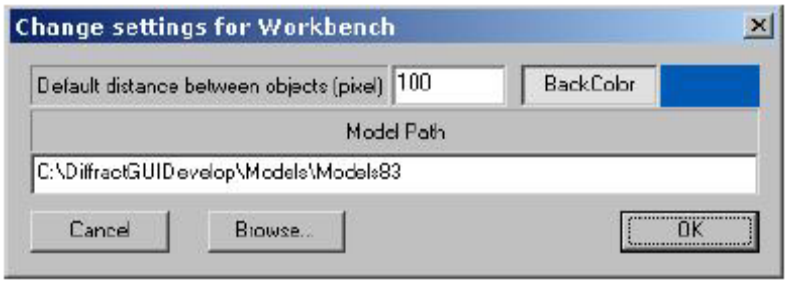
矢印キー（←,↑,→,↓）を使うと、object を 10pixel だけ平行移動できる。また、**Ctrl** キーを押しながら矢印キーを押すと、object は 10°だけ回転する。このとき、もし 1 個の object を選択すると、その object だけが平行移動・回転し、選択しないでこの処理を行うと全部の object が平行移動・回転を行う。

Menu から **Edit-Inflation** および **Edit-Deflate** を使うと、選択した object（あるいは特定の object を選定しないときは、object のグループ全体）のサイズを大きくしたり、小さくしたりできる。また **Edit-Normal Size** を選ぶと、すべての object は model ファイルで決められているオリジナルのサイズに再描画される。

もし、システムの全体のサイズがディスプレイ面の大きさを超えるときは、Menu から **Edit-FitToScreen** を選べば、面の大きさに収まるように調節される。他の方法は、スクロールバーを選べばよい。スクロールバーはスクリーンの右および下にあり、object または object のグループがディスプレイ面の左側または上で切れている時は、**Edit-AlignToTop** を使って object を画面内に戻せばよい。（これらの操作のすべては、ディスプレイ面内のみで作用し、COMMAND.DAT への影響はない。）

5.2　Workbench の option (Options for Workbench)

Menu:**View-Setting-View3D** によって、次のような対話ボックスが起動される。



model パスは model ファイルが存在しているフォルダーを指定しなければならない。そう

しなければ *DiffractGUI* は COMMAND.COM の内容を読むことはできない。表示された 3Dobject 間のデフォルト距離は（見やすくするために）変えることができるが、それは常にゼロより大きくなければならない。OK ボタンをクリックするとこれらの option は Windows のレジストリに保存され、その後の *DiffractGUI* セッションでも有効となる。

5.3 新しい object の導入(Create a new object)

新しい object を作り、既存のシステムに挿入するには、下に述べるように 2 つの方法がある。どちらの方法も任意のタイプの object を、選択した object の後側に挿入することができる。もし、object を選択しなかった場合は、新しい object は既存の一連の object の最後に挿入される。最初は object の parameter はデフォルト値に設定するが、新しい object が挿入され表示されたら、そのパラメーターを **Edit-DIFFRACT-paraeters** 対話ボックスを使って変更することができる。

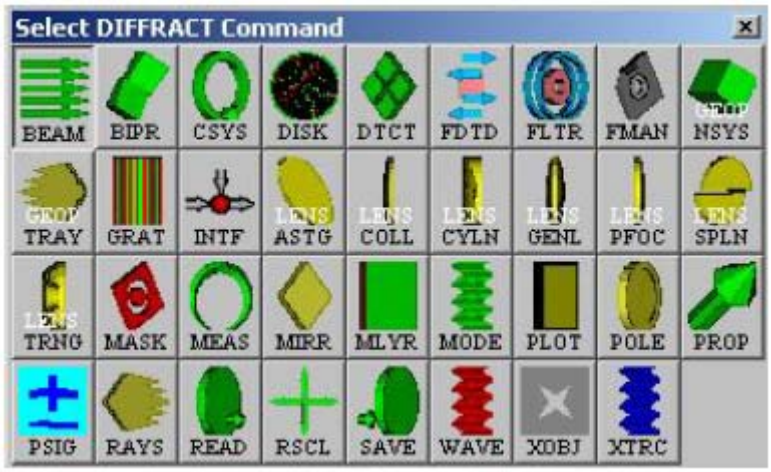
5.3.1 ChooseObject 対話ボックス（ChooseObject dialog）

ChooseObject 対話ボックスは、menu から **Edit-Create** を選択すれば開かれる。リストに表示されたモデル（それぞれの model は DIFFRACT の menu 命令に対応している）の一つを選択して OK をクリックすると、新しい object が作られ、システムに挿入することができる。

5.3.2 Command Panel

Command Panel は希望する icon をマウスでクリックすることによって新しい object を作る手取り早い方法である。Menu から **View-CommandPanel** を選ぶと Command Panel を

使う（または使えないようにする）ことが出来るようになる。Command Panel を始めて使う時は、そのアイコンが形成され、Comand Panel が最終的な形になるまでに数秒かかる。Command Panel は **Workbench** あるいは **ScriptView** から使うことが出来る。ディスプレイされたアイコンの数は、Model フォルダーの中の model ファイルの数によってきまる。それぞれのアイコンの特性は、対応する model ファイルに記録されている情報によって決まる。



5.4　object の削除（Delete an Object）

object を削除する時は、当該 object を選択し、キーボードの Delete キーを押すか、menu から **Edit-Cut** を選択すればよい。もし、object を選択しないときは ChooseObject 対話ボックスが現れ、ユーザーに削除する object を決めるように促す。

5.5 切り取り、コピー、 貼り付け（Cut, Copy, Paste）

Workbench の中で、Edit-Copy を使えば、選択した object を Windows の Clipboard へコピーすることができる。Clipboard の内容は上書きされる。**Edit-Cut** によって object を削除する時、その object は一旦 clipboard に送られる。したがって、不用意に削除した場合、その object は **Edit-Paste** によって回復することができる。**Edit-Paste** は新しい object（あるいは一連の object）を clipboard から取り込むことができる。もし、選択した object が Workbench に存在した場合、新しい object は選択した object の直後に挿入される。もし、object を選択しないときは、新しい object は既存の object line の後に添付される。
複数の object は clipboard の中ではテキスト行のように扱われる。これは DIFFRACT の command block の import または export は、任意の外部テキストエディターに対して実行することができることを意味している。Clipboard から貼り付けるときは、DiffractGUI は clipboard のテキストの内容をチェックする。第 1 行の最初の字が”!”または”$”でないときは、

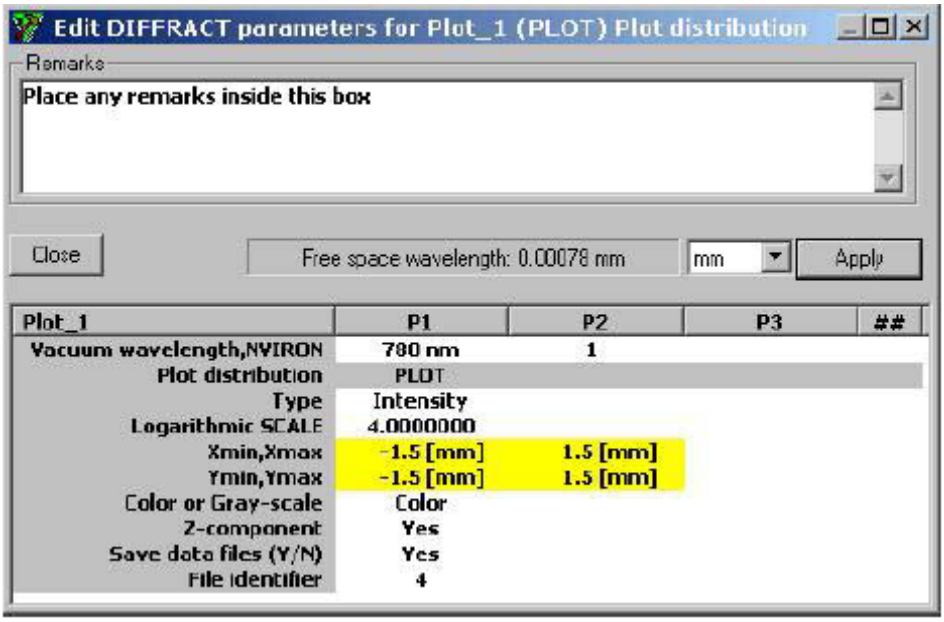
clipboard の内容はエラーであると判断して paste を拒否する。これは、外部のテキストエディターから新しい object を取り入れるときに重要なことであって、単に copy した第 1 行目の最初の文字が"!"または"$"であることを確かめ、それが no の場合には DiffractGUI は paste を実行しない

5.6 Object の順序の変更（Change the order of objects）

Object の配列順序は ChooseObject- d ialog で変更することができる。ChooseObject-dialog を実行する最も簡単な方法は、マウスで background をダブルクリックするか、object を指定しないで menu から **DIFFRACT-Edit Parameters** を選択すればよい。この dialog を開いた後、ディスプレイリストから object を選択し、Shift Up または Shift Down キーを使ってその object をリスト内で動かせばよい。もう 1 つのやり方は cut & paste を使う方法である。ある object を元の位置で cut し、新しい位置で paste すればよい。

5.7 Object の DIFFRACT パラメーターの変更（Change DIFFRACT- parameters of an object）

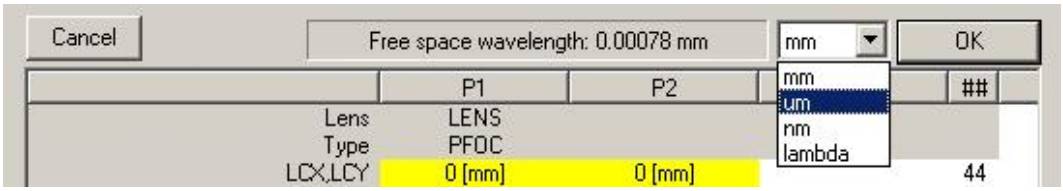
この操作は **Edit DIFFRAC parameters** 対話ボックスで実行できる。そのためには、目的の object をダブルクリックするか、menu から **DIFFRACT-Edit parameters** を選べばよい。リスト box の内容は選んだ object の model ファイルから得られる情報によって構成されているため、その object の種類によって異なる。下に示す例は、LENS, POFC タイプの対話ボックスであって、上欄は COMMAND.DAT の comments に似た guiding 情報である。下欄は対話ボックスで、COMMAND.DAT ファイルの 1 行は 3 個のパラメーターと optional label number (10~99)から構成されているため、このボックスにも 4 個の列, P1, P2, P3, ## がある。



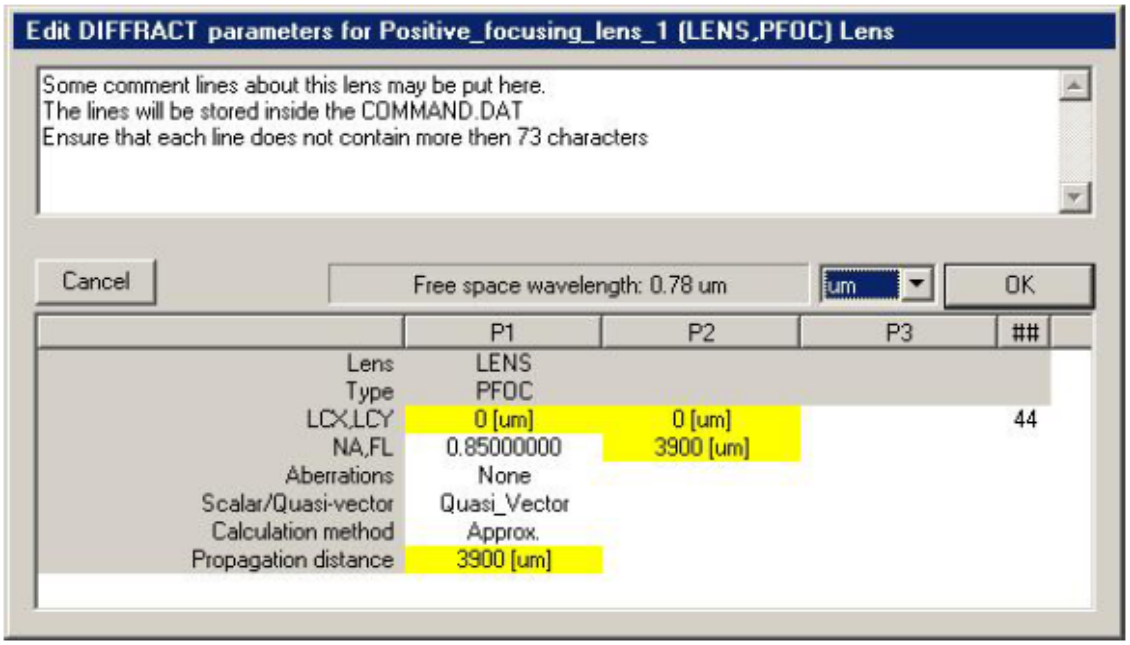
黄色で強調されているパラメーターは長さを表すものであって、units-combo-box で指定された単位で表示されている。（Version 7.7, 7.9, 8.1 では、DIFFRACT で処理される前に真空中の波長で基準化される。この変換はセッションの開始時、または COMMAND.DAT ファイルの先頭で設定された波長を使う。）この dialog box の上部にある edit box において、編集中の object に対する任意の行数の comment line を付与することができる。これらの comment line は COMMAND.DAT ファイルに保存され、当該 object のテキストブロックの前に置かれる。たとえば、上記の例では COMMAND.DAT では次のように表されている。

```
!Some comment lines about this lens may be put here.                    #
!The lines will be stored inside the COMMAND.DAT                        #
!Ensure that each line does not contain more then 73 characters        #
$                                                                       #
                  Lens: LENS                                (Units=lambda)  #
                  Type: PFOC                                              #
               LCX,LCY:  0.00000000  0.00000000                           #44
                 NA,FL:  0.85000000  5000.0000                            #
           Aberrations: None                                            #
  Scalar/Quasi-vector: QV                                               #
   Calculation method: APRX                                             #
  Propagation distance:  5000.0000                                      #
$                                                                       #
```

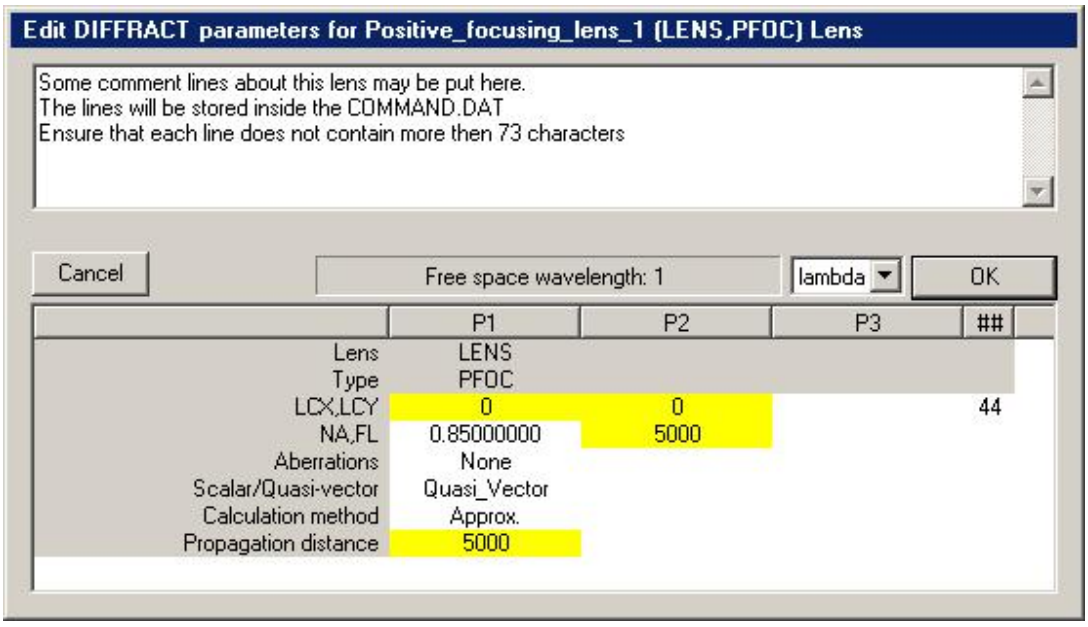
長さの単位は unit-combo-box で変更することができる。



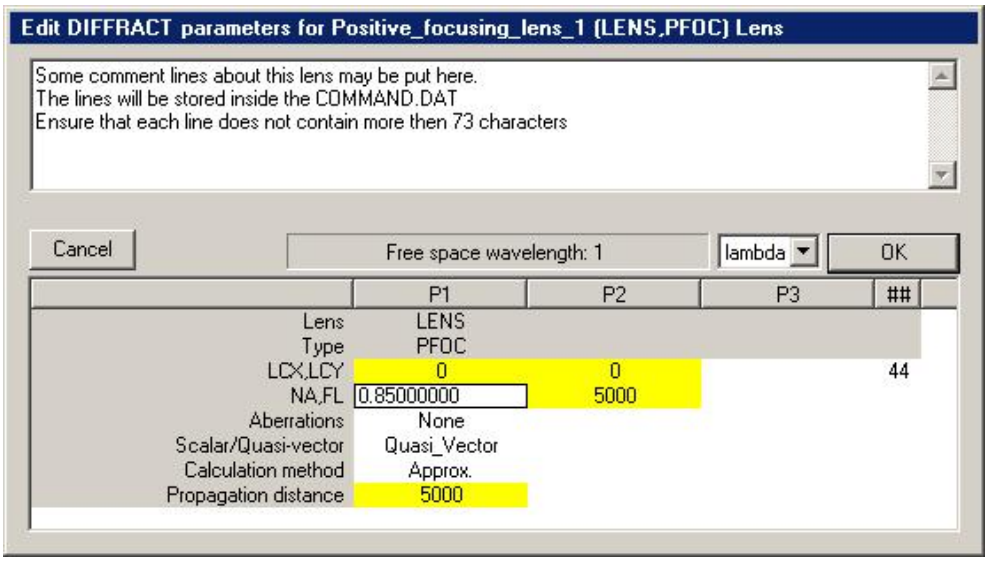
上の選択によって次のようになった。



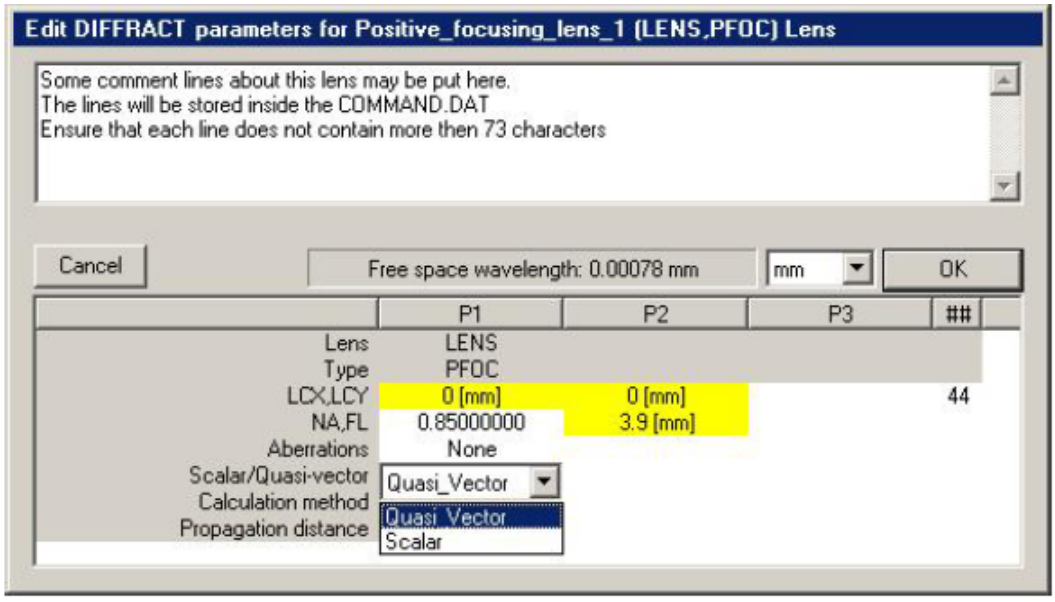
長さの値はこの選択では変わらず、単に表示された数値と単位[mm]が[um]に変わっただけである。単位として[lambda]を選択すると、すべての長さが自由空間の波長で基準化されて表示される。



この場合には長さの数値には（[lambda]のような）単位表記はないが、数値は長さであることを記憶するために黄色で強調された形で残ることに注意されたい。さらに、自由空間の波長を表す数値は 1 に変わり、それは長さに対する単位として使われたと表示することに注意されたい。

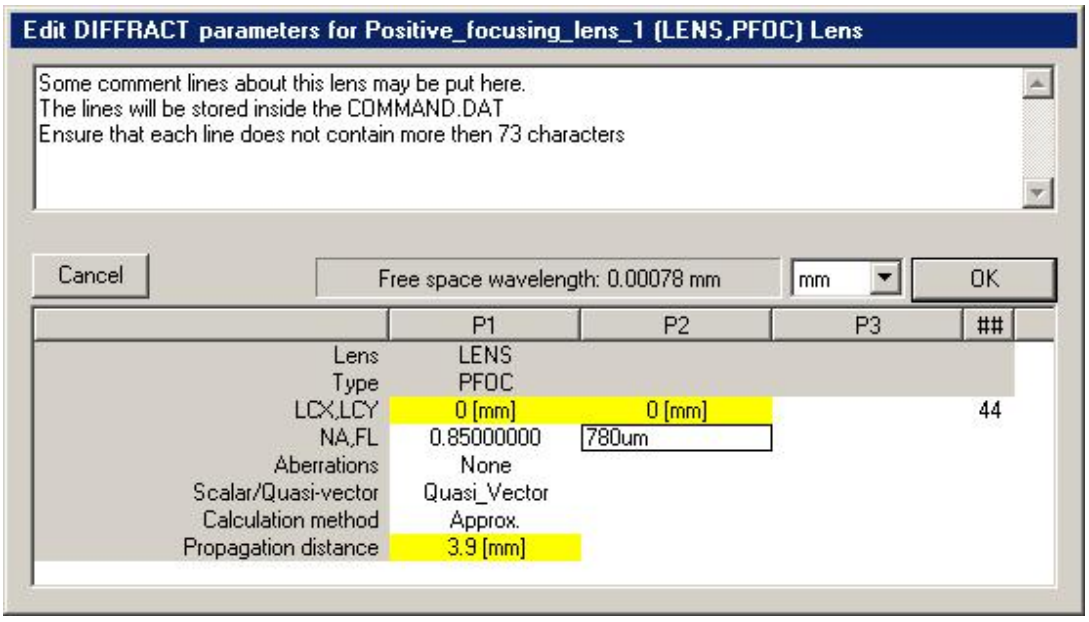


パラメーターの数値またはテキストに対して左マウスをクリックすると、edit box(上を見よ)、あるいは combo box(下の図を見よ)が開いて変更が可能となる。Edit box や combo box はリストボックスの内部の他のパラメーターを左クリックすれば閉じることができる。現れる box のタイプはパラメーターのタイプによって異なり、数値またはファイル名は edit box、２～３の option を行うパラメーターは combo box を使う。Combo box のリストは拡張可能で、選択したパラメーターに対してすべての可能な option を使うことが出来る。Edit box はもし対応する値が数値であるなら単に数字（整数と浮遊小数点）だけを受付け、filename-edit-box は単に正しいファイル名だけを受け付ける（すなわち、4 文字名と適当な拡張子)。

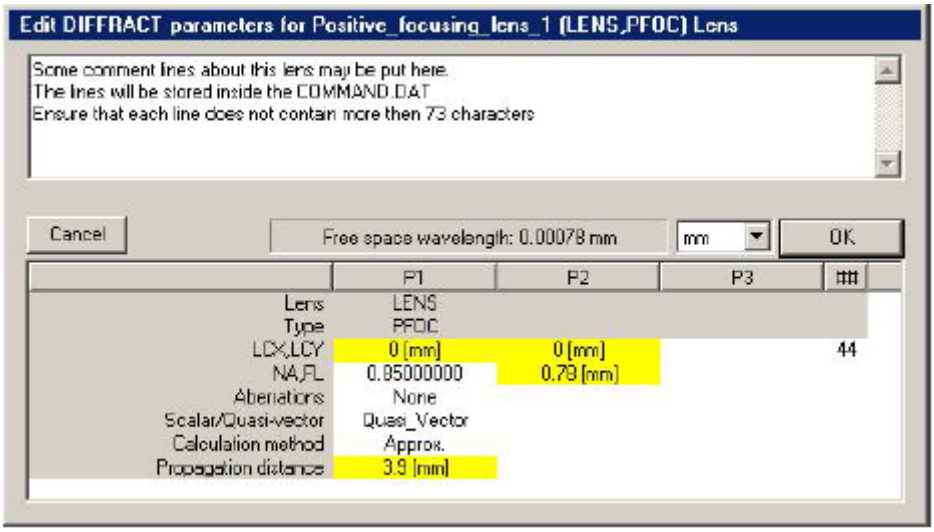


パラメーターの編集が終われば、ENTER, TAB, Shift+TAB, あるいはパラメーターの他の部分を押せばよい。もし、TAB(あるいは Shift+TAB)をつかうと、次の（または前の）パラメーターが自動的に開かれ、編集が可能となる。

もし、編集したパラメーターが長さであると、suffix ”mm”, ”um”あるいは”nm”を数値に付けて、その長さの単位をきめることが出来る。1 例としてパラメーターFL(焦点距離 focal length)を上の図に 780um と設定して見よう。それには単に次のようにタイプして Enter を押せばよい。その結果



Enter を押せばよい。その結果



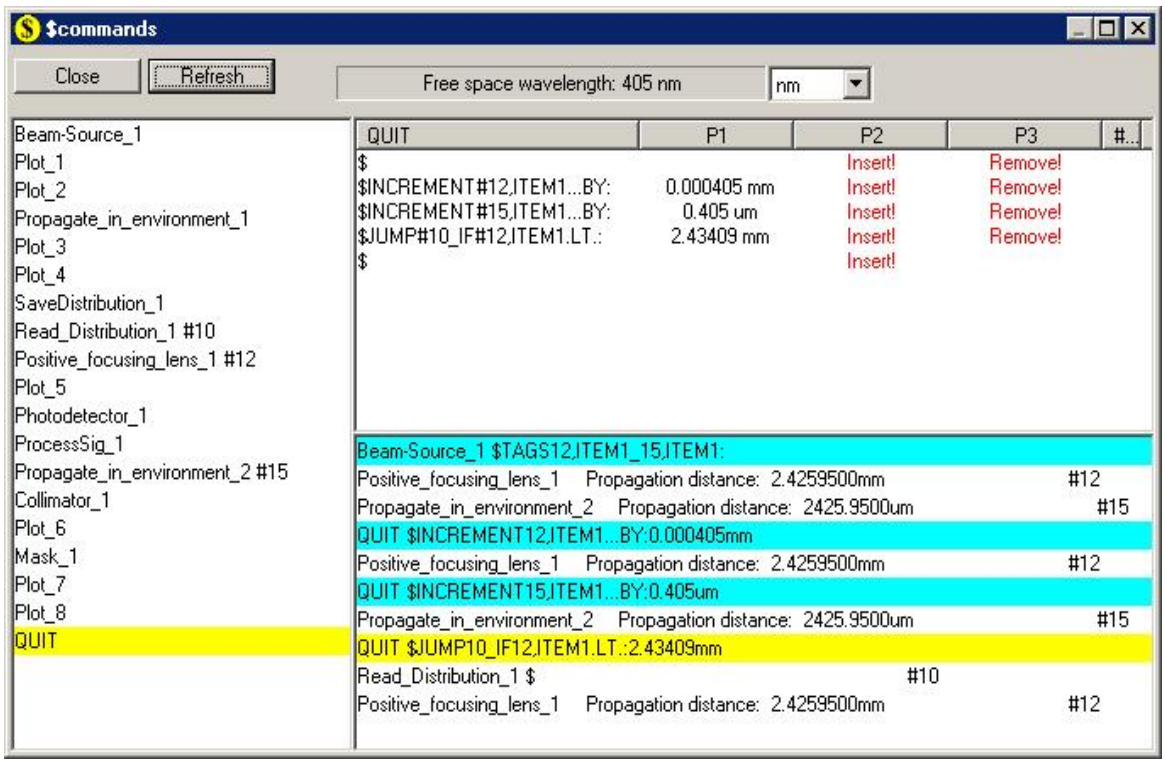
となる。もし suffix “mm”, “um”あるいは”nm”を省略すると、*DiffractGUI* は unit-combo-box によって特定された単位であると判断する。（上の場合は、単位は mm と考える。）

この対話ボックスの便利な特色は、サイズを変えることができることである。サイズを変えるには、マウスカーソルを dialog window の縁に置き、マウスの形が変わるのを待って希望のサイズにドラッグするとよい。変更した値を保存するためには、Dialog box は”OK”で（または ENTER を押して）閉じなければならない。もし”Cancel”で閉じると、すべての変更は放棄される。

註：DiffractGUI はユーザーによって入力された object のパラメーターの値をチェックしない。たとえば、レンズの開口数として−137 を入力したとすると、これは間違いなく意味のない数値であるが、DiffractGUI は受け入れてしまう。ユーザーは正しいパラメーター値を入力する責任を負う。

5.8　$命令エディター(The $-commands editor)

DIFFRACT の $ 命令は、非常に柔軟性があり、非常に簡単な構文を持っているのが、重要な機能である。DiffractGUI の**$-command editor** は command.dat ファイル中に$command を挿入したり、あるいは操作することができる。**$-command editor** は ToolBar の をクリックするか、または menu から **DIFFRACT-$commands** で起動すればよい。



上の図で分かるように、$command editor は 3 個のリストを持っている。Window の左側にあるリスト、すなわち **object-list** は、(command.dat ファイルの command block に対応して) **Workbench** 上にあるすべての object を表示している。リストの最後には QUIT という"virtual" object があるが、これはファイルの終わりの直前にある最後の command block の後におかれた$command を特定するためである。それぞれの object には、対応する command block の中に存在するすべてのラベルのリストが付けられている。

Object-list の中からどれかの object を選ぶと、その object に対応する command block が dialog box の右上の窓に表示される。これは EDP-list (Edit DIFFRACT Parameter の略語) に相当する。EDP-list は、**Edit DIFFRACT Parameters** の dialog box であって、必要な command block とすべての関連する$command を表示し、その挿入、削除、変更などすべての編集が可能となる。

右下の窓は**$commands editor** window であって、その中の commands-list は今考えている command.dat ファイル中のすべての$command を表示している。それぞれの$command はその$command の(ラベルを付けた)target line を示した 1−2 行のラインの後に書かれている。この command-list からある行を選ぶと、その行に関連する object が object-list 内で自動的にマークが付けられ、そのパラメーターが EDP-list に表示される。
EDP-list は**$command editor** で変更できる唯一の list であって、*Object-list* と *command-*

list は全体を眺めたり、情報を得るためのものであって、編集はできない。 *EDP-list* で Insert!ラベルをクリックすれば空の$-line を挿入することができる。逆に、$line を削除するためには関連の Remove!ラベルをクリックすればよい。（各 command block は少なくとも 1 つの$-line を経て進まなければならないので、最後の$-line は削除できない。

COMMAND,DAT ファイルを読み込んでいるときに、$-line のない command block が見つかると、DiffractGUI は自動的に空の$-line を挿入する。）Remove!をクリックすると、ユーザーに警告なしに、直ちに$-line が削除されることに注意されたい。したがって、*EDP-list* でのクリック操作には注意が必要である。

$-line の内容は、EDPlist で当該 line をクリックすると、他のパラメーターと同様に、変更することができる。Edit box は、キーボードからの入力があると、その line を入れ替える。完全な$command をタイプする必要はなく、最初の 1～2 字をタイプすれば充分である。（また、最初の"$"もタイプする必要はない。）たとえば、もし$TAGS#12,ITEM1_15,ITEM1 と入力したい時は、"t12,1,15,1"とタイプして ENTER を押せば、*DiffractGUI* は残りを処理してくれる。もし、データが欠落している時には、挿入された $ command 中に"?"が挿入される。

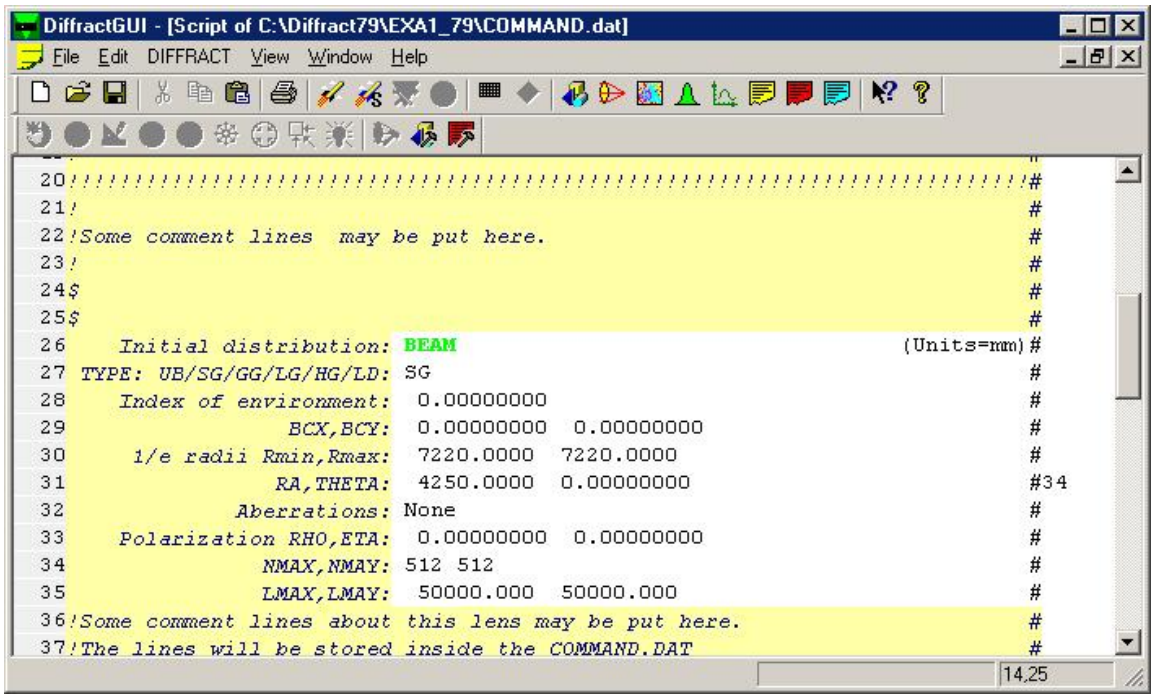
ある$command（たとえば、INCREMENT や JUMP）はコロン（：）の後に制御値が必要である。この値は他のパラメーターと同様に編集することができる。この値には単位が必要であるが、$command の labeled target line で用いられた長さの単位は、その固有の単位（すなわち, mm, um, nm）に従わなければならない。command block で用いた単位にのみ依存していることに注意されたい。もし、target parameter が長さではない場合は、$command は単位を必要としない。

$-line は *EDP-list* の中では drag&drop で移動することができる。動かしたい line にマウスカーソルを置き、右ボタンを押しながら希望の場所まで drag し、ボタンを離せばよい。*EDP-list* における変更が終わったら、**Reflesh** ボタンをクリックするか、*Enter* を押すことが重要である。このようにすれば、*object-list* と *command-list* は変更を受け入れる。もしエラー（ラベルの重複とか、欠如など）が起こると、*command-list* に表示される。

6. Script の編集（ScriptView）

ScriptView は文書（すなわち COMMAND,DAT）の内容を DIFFRACT ユーザーが読めるように表示するテキストエディターである。また、この文書の編集も可能である。
ScriptView の background では ScriptWizard というルーチンが起動し、ユーザーがタイプ

した構文のエラーを検出し、訂正する。


DiffractGUI - [Script of C:\Diffract79\EXA1_79\COMMAND.dat]
File Edit DIFFRACT View Window Help
20 !!!!

註：DIFFRACT ヴぇr。7.7、7.9、および8.1では、ScriptViewに現れるすべての長さは真空中の波長で基準化されているので、テキストのブロック"(Unit=xx)"は、ここでは重要ではなく、ユーザーは無視するかもしれない。しかし、Edit-DIFFRACT_Parameterを行うとき、このtext blockはユーザーが初めに決めた単位で長さのパラメーターを表示する。

6.1 ScriptWizard

ScriptWizard は COMMAND.DAT インタープリターであって、text caret[†]が line 間で移動したり、text 変更を見つけた時自動的に起動する。Caret が同一 line 内に留まる場合は ScriptWizard によらずに 1 line 内の修正を何回でも実施できる。

ScriptWizard が起動すると、テキストを解釈し、すべての有効な変更を document に転送する。また、必要に応じて、command block を削除したり、新設したりする。もしテキストがエラーを含んでいる場合は、ScriptWizard は対応する command block を document から復元する。Comment area を形成する line の最初の 25 文字は、ScriptWizard で無視される。ただし、最初の文字が"$"である場合は例外で、対応する$command を読み込み、解釈する。

6.2 DIFFRACT menu command block の新設 (Creating a new DIFFRACT menu command block)

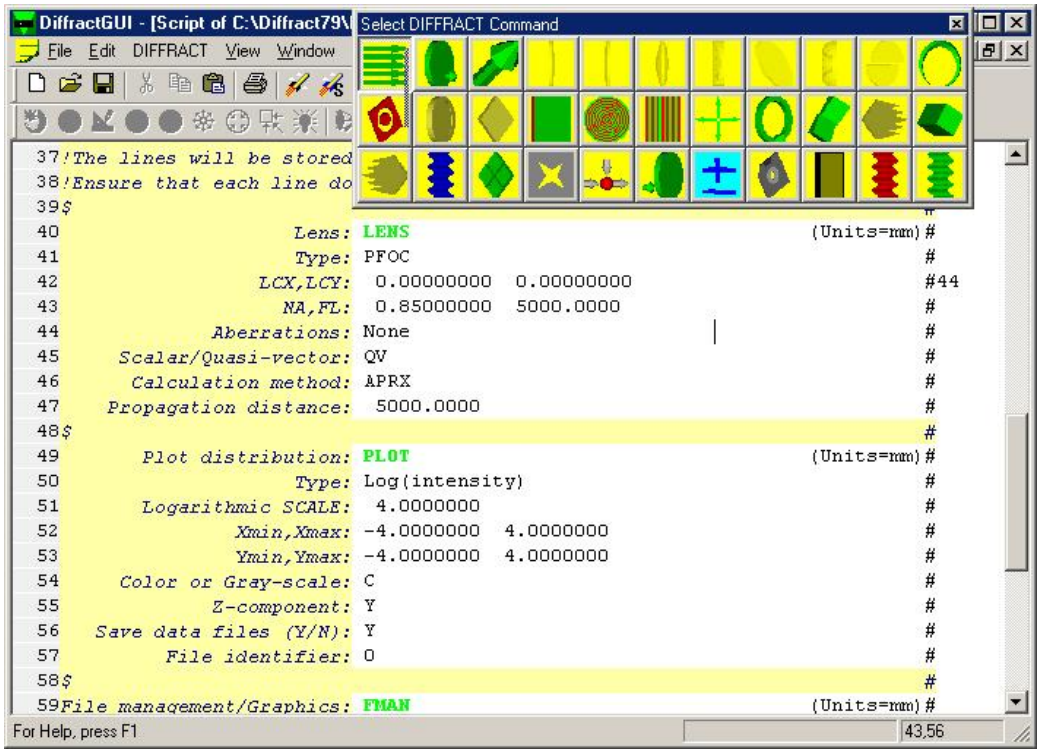
6.2.1 Using the CommandPanel

CommandPanel は希望する object の上でマウスをクリックすることによって新しい object を作る手軽な方法を提供する。Menu から View-CommandPanel を選んで起動できる。CommandPanel の最初の drawing では、アイコンが作られ、CommandPanel が完全な形になるのに数秒かかる。CommandPanel は View3D と Script View から起動できる。CommandPanel からアイコンを選ぶと、Text caret を含む block のすぐ後に、新しい command block が document の中に作られる。新しい block のパラメーターは対応する model ファイルからデフォルト値で初期化できる。

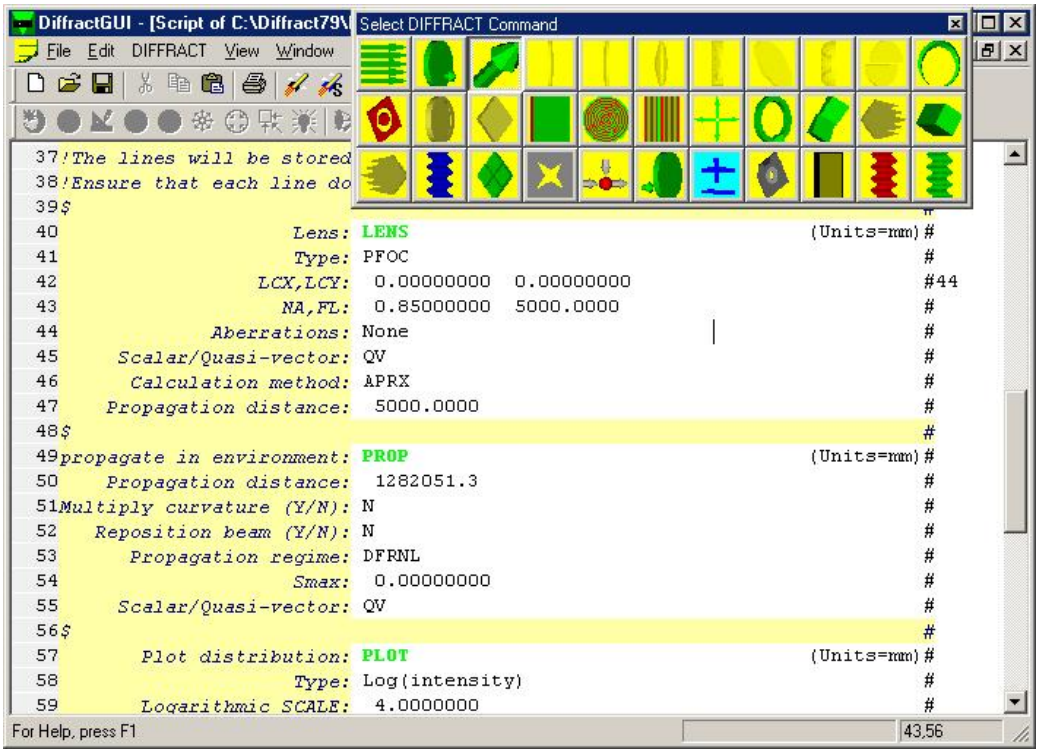
例: 下の図を考え、LENS と FMAN block の間に PROP を挿入する。これを実行するには、text caret を line 41 から 47 の間に置き、(ToolBar にある をクリックして)

† カーソルで表示される校正記号。"|"の形となり校正する位置を示す。

CommandPanel を開く。


DiffractGUI - [Script of C:\Diffract79\
File Edit DIFFRACT View Window
Select DIFFRACT Command
37!The lines will be stored
38!Ensure that each line do
39$
40 Lens: LENS (Units=mm) #
41 Type: PFOC #
42 LCX,LCY: 0.00000000 0.00000000 #44
43 NA,FL: 0.85000000 5000.0000 #
44 Aberrations: None #
45 Scalar/Quasi-vector: QV #
46 Calculation method: APRX #
47 Propagation distance: 5000.0000 #
48$ #
49 Plot distribution: PLOT (Units=mm) #
50 Type: Log(intensity) #
51 Logarithmic SCALE: 4.0000000 #
52 Xmin,Xmax: -4.0000000 4.0000000 #
53 Ymin,Ymax: -4.0000000 4.0000000 #
54 Color or Gray-scale: C #
55 Z-component: Y #
56 Save data files (Y/N): Y #
57 File identifier: 0 #
58$ #
59File management/Graphics: FMAN (Units=mm) #
For Help, press F1
43,56

CommandPanel の PROP アイコンをクリックすれば次のようになる。

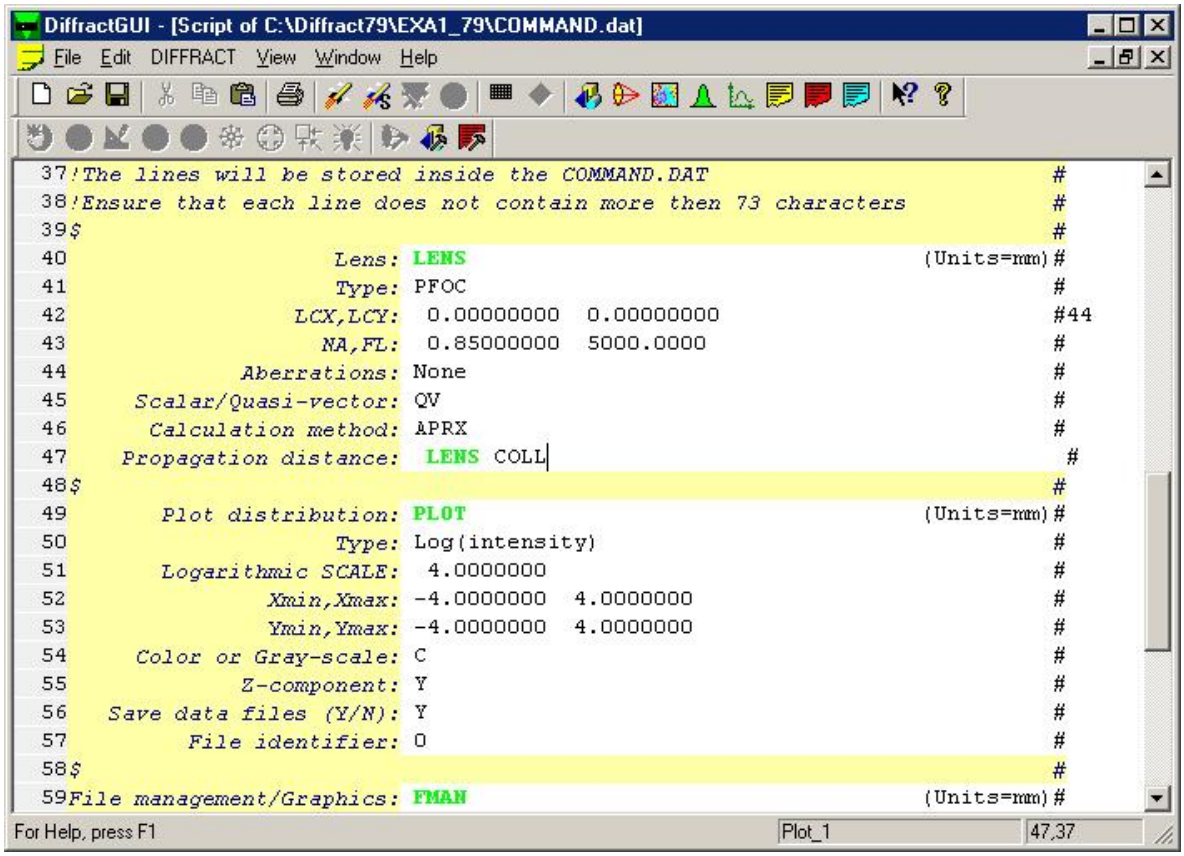

DiffractGUI - [Script of C:\Diffract79\
File Edit DIFFRACT View Window
Select DIFFRACT Command
37!The lines will be stored
38!Ensure that each line do
39$
40 Lens: LENS (Units=mm) #
41 Type: PFOC #
42 LCX,LCY: 0.00000000 0.00000000 #44
43 NA,FL: 0.85000000 5000.0000 #
44 Aberrations: None #
45 Scalar/Quasi-vector: QV #
46 Calculation method: APRX #
47 Propagation distance: 5000.0000 #
48$ #
49propagate in environment: PROP (Units=mm) #
50 Propagation distance: 1282051.3 #
51Multiply curvature (Y/N): N #
52 Reposition beam (Y/N): N #
53 Propagation regime: DFRNL #
54 Smax: 0.00000000 #
55 Scalar/Quasi-vector: QV #
56$ #
57 Plot distribution: PLOT (Units=mm) #
58 Type: Log(intensity) #
59 Logarithmic SCALE: 4.0000000 #
For Help, press F1
43,56

この後、挿入した command block のパラメーター値を訂正できるようになる。

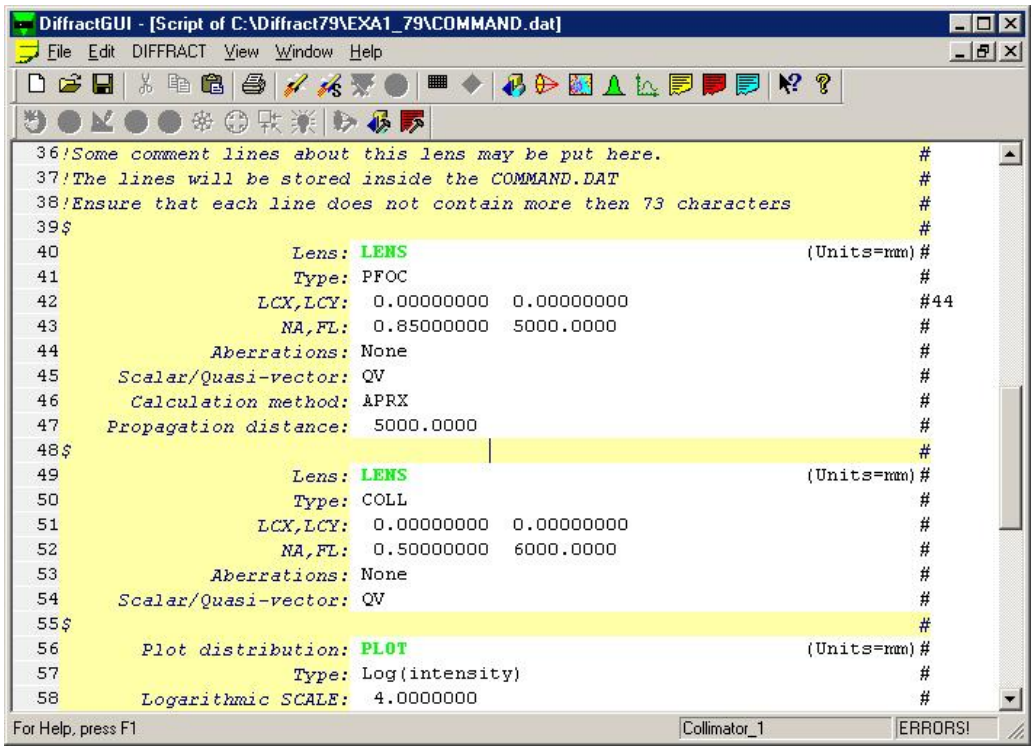
6.2.2　キーボードからのタイピング（Typing from the keyboard）

キーボードによって新しい DIFFRACT menu command block を作るには、該当する keyword を parameter line の non-comment area で、既存の parameter に上書きしてタイプしなければならない。Caret が他のラインに動くと直ちに ScriptWizard が作動して、上書きされた line を修正し、今入力したキーボードによる新しい command block を挿入する。この新しい block は修正したラインに添加される。もし、keyword が（unique identification のために）subkey を必要とする場合は、その subkey を keyword に続いてテキストの同じ line に、少なくとも keyword の 1 字分のブランクをおいて、入力しなければならない（たとえば　LENS PFOC または LENS COLL のように）。

例：ここで、前のセクションで述べた状態を考え、正の焦点距離をもつ LENS と FMAN ブロックの間にコリメーターLENS を挿入する。これを実行するためには、positive focus LENS の最後のパラメーター（line 47）を LENS COLL と置き替え、


DiffractGUI - [Script of C:\Diffract79\EXA1_79\COMMAND.dat]
File Edit DIFFRACT View Window Help
37 !The lines will be stored inside the COMMAND.DAT #
38 !Ensure that each line does not contain more then 73 characters #
39 $ #
40 Lens: LENS (Units=mm) #
41 Type: PFOC #
42 LCX,LCY: 0.00000000 0.00000000 #44
43 NA,FL: 0.85000000 5000.0000 #
44 Aberrations: None #
45 Scalar/Quasi-vector: QV #
46 Calculation method: APRX #
47 Propagation distance: LENS COLL #
48 $ #
49 Plot distribution: PLOT (Units=mm) #
50 Type: Log(intensity) #
51 Logarithmic SCALE: 4.0000000 #
52 Xmin,Xmax: -4.0000000 4.0000000 #
53 Ymin,Ymax: -4.0000000 4.0000000 #
54 Color or Gray-scale: C #
55 Z-component: Y #
56 Save data files (Y/N): Y #
57 File identifier: 0 #
58 $ #
59 File management/Graphics: FMAN (Units=mm) #
For Help, press F1
Plot_1
47,37

そしてカーソルを line 48 に動かすため、”↓”キーを押せばよい。結果は次のようになる。

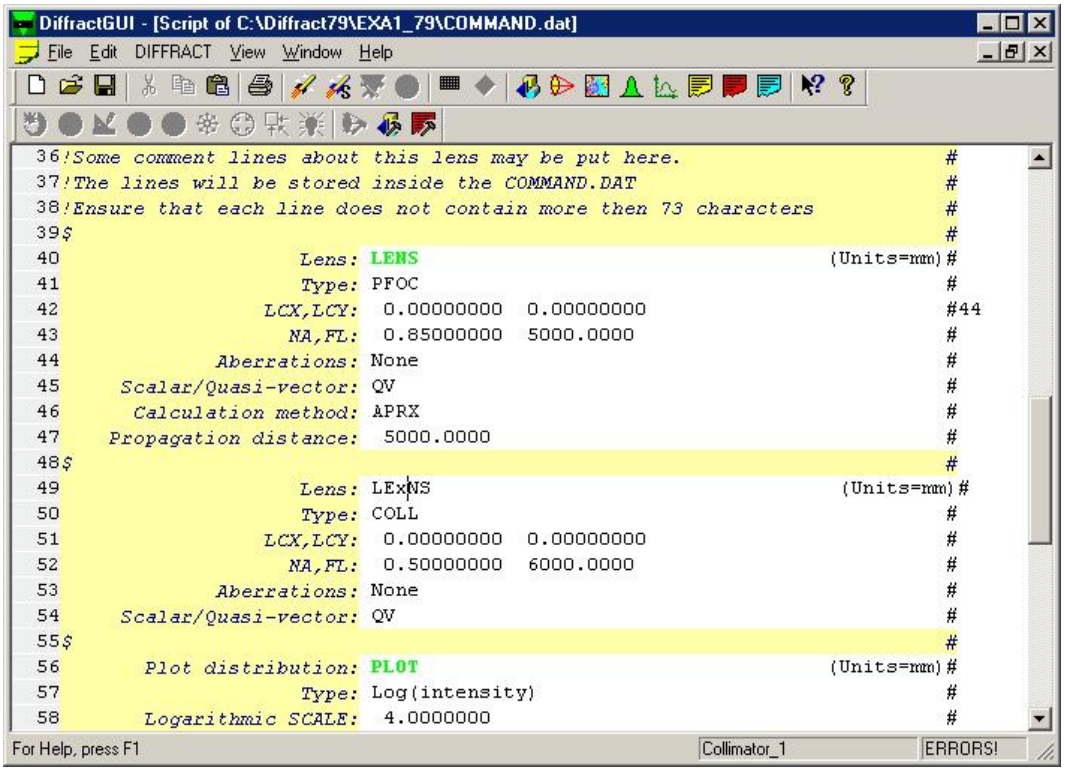

DiffractGUI - [Script of C:\Diffract79\EXA1_79\COMMAND.dat]
File Edit DIFFRACT View Window Help
!Some comment lines about this lens may be put here. #
!The lines will be stored inside the COMMAND.DAT #
!Ensure that each line does not contain more then 73 characters #
$ #
Lens: LENS (Units=mm) #
Type: PFOC #
LCX,LCY: 0.00000000 0.00000000 #44
NA,FL: 0.85000000 5000.0000 #
Aberrations: None #
Scalar/Quasi-vector: QV #
Calculation method: APRX #
Propagation distance: 5000.0000 #
$ #
Lens: LENS (Units=mm) #
Type: COLL #
LCX,LCY: 0.00000000 0.00000000 #
NA,FL: 0.50000000 6000.0000 #
Aberrations: None #
Scalar/Quasi-vector: QV #
$ #
Plot distribution: PLOT (Units=mm) #
Type: Log(intensity) #
Logarithmic SCALE: 4.0000000 #
For Help, press F1
Collimator_1
ERRORS!

再び、ユーザーは挿入した command block に付属している default パラメーターを変更するように作業を続けるとよい。

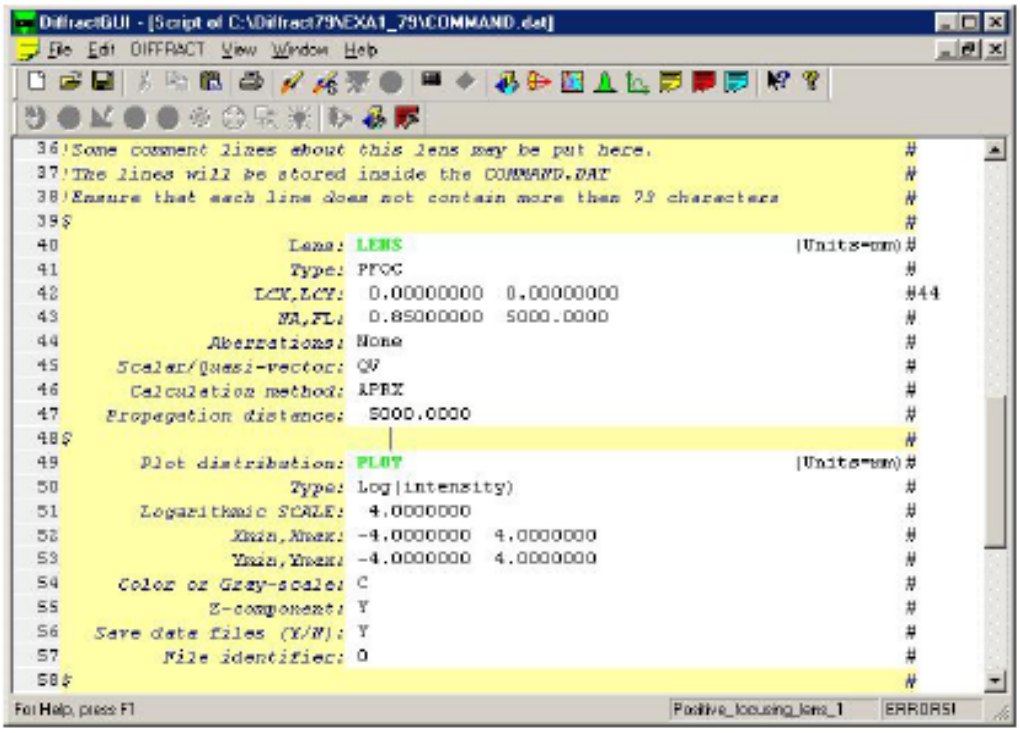
6.3 DIFFRACT menu の command block の削除 (Deleting a DIFFRACT menu command block)

Command block はその key word を定義されていない keyword に変え（たとえば BEAM を BexAM に変える）て、caret を他の行に移すことによって削除することができる。こうすれば、ScriptWizard は全体の command block を定義されない keyword と一緒に削除する。

例 : LENS COLL を削除しよう。上に述べたように、LENS をたとえば LExNS に変える。すると ScriptWizard は LexNS を認識できないので、その word の色を変える。


DiffractGUI - [Script of C:\Diffract79\EXA1_79\COMMAND.dat]
File Edit DIFFRACT View Window Help
36 !Some comment lines about this lens may be put here. #
37 !The lines will be stored inside the COMMAND.DAT #
38 !Ensure that each line does not contain more then 73 characters #
39 $ #
40 Lens: LENS (Units=mm) #
41 Type: PFOC #
42 LCX,LCY: 0.00000000 0.00000000 #44
43 NA,FL: 0.85000000 5000.0000 #
44 Aberrations: None #
45 Scalar/Quasi-vector: QV #
46 Calculation method: APRX #
47 Propagation distance: 5000.0000 #
48 $ #
49 Lens: LExNS (Units=mm) #
50 Type: COLL #
51 LCX,LCY: 0.00000000 0.00000000 #
52 NA,FL: 0.50000000 6000.0000 #
53 Aberrations: None #
54 Scalar/Quasi-vector: QV #
55 $ #
56 Plot distribution: PLOT (Units=mm) #
57 Type: Log(intensity) #
58 Logarithmic SCALE: 4.0000000 #
For Help, press F1
Collimator_1
ERRORS!

ここで text caret を line 49 から外すと、LENS COLL は script から削除される。


DiffractGUI - [Script of C:\Diffract79\EXA1_79\COMMAND.dat]
File Edit DIFFRACT View Window Help
36 !Some comment lines about this lens may be put here. #
37 !The lines will be stored inside the COMMAND.DAT #
38 !Ensure that each line does not contain more then 73 characters #
39 $ #
40 Lens: LENS (Units=mm) #
41 Type: PFOC #
42 LCX,LCY: 0.00000000 0.00000000 #44
43 NA,FL: 0.85000000 5000.0000 #
44 Aberrations: None #
45 Scalar/Quasi-vector: QV #
46 Calculation method: APRX #
47 Propagation distance: 5000.0000 #
48 $ #
49 Plot distribution: PLOT (Units=mm) #
50 Type: Log(intensity) #
51 Logarithmic SCALE: 4.0000000 #
52 Xmin,Xmax: -4.0000000 4.0000000 #
53 Ymin,Ymax: -4.0000000 4.0000000 #
54 Color or Gray-scale: C #
55 Z-component: Y #
56 Save data files (Y/N): Y #
57 File identifier: 0 #
58 $ #
For Help, press F1
Positive_focusing_lens_1
ERRORS!

6.4 DIFFRACT menu command block のパラメーター値の変更（Changing parameter values of DIFFRACT menu command blocks）

キーボードで **TAB**（または **SHIFT+TAB** Key）を押すと、caret は（下降または上昇で）１つのパラメーターの値から次の値に、テキストを通り越して下方（または上方）に移る。更に、最も近い値が選択され、変更が容易に行われる。Text caret が他の line に移ると、ScriptWizard は変更された line をチェックし、解釈する。もし、変化が DIFFRACT のルールに適合し、値が理解可能であると、ScriptWizard はそれを受け入れるが、そうではないと有効な値を含む前の command block を復元する。このようなことが起こると、LoggerViewer を開いて、なぜ ScriptWizard が変更を拒否したかという理由を調べた方が良い。

Version 7.7, 7.9 および 8.1 では、すべての長さは波長の単位で表示され、その COMMAND,DAT の正しい内容を反映している。しかし、取り扱いを容易にするため、ScriptWizard は command line を解釈するときに、その長さの単位（mm, um, nm）を認識する。

例：LENS PFOC の焦点距離を 1.5ｍｍに設定したい。そのためには、パラメーターの値として、単に 1.5ｍｍとタイプ（1.5 と mm の間にはブランクを入れない）する：

```
39$                                                                    #
40                    Lens: LENS                              (Units=mm)#
41                    Type: PFOC                                       #
42                  LCX,LCY:  0.00000000  0.00000000                   #44
43                   NA,FL:  0.85000000  1.5mm                    #
44              Aberrations: None                                      #
45      Scalar/Quasi-vector: QV                                        #
46       Calculation method: APRX                                      #
47     Propagation distance:  5000.0000                                #
```

そこで、text caret を別の line に移すと、ScriptWizard は調節を行い、その結果は次のようになる。

```
39$                                                                    #
40                    Lens: LENS                              (Units=mm)#
41                    Type: PFOC                                       #
42                  LCX,LCY:  0.00000000  0.00000000                   #44
43                   NA,FL:  0.85000000  1923.0769                     #
44              Aberrations: None                                      #
45      Scalar/Quasi-vector: QV                                        #
46       Calculation method: APRX                                      #
47     Propagation distance:  5000.0000                                #
48$                                                                    #
```

1.5mm の長さは自由空間の波長で基準化されている。上の例では、自由空間の波長は 780nm である。”1.5ｍｍ”の代わりに”1500μm”とか”1.5e6nm”とタイプしても同じ結果が得られる。

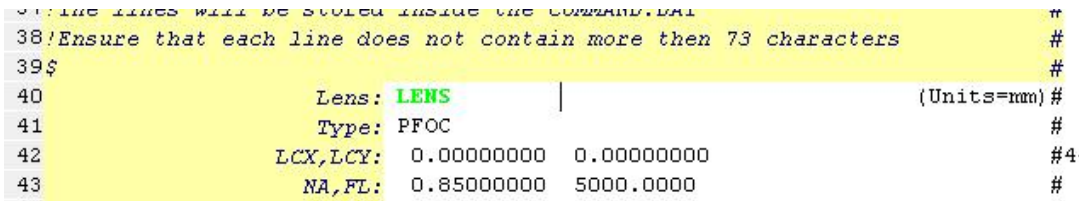
6.5 $命令（$commands）

註：$command に対しては、$command-editor の方が ScriptView よりも更に使いやすい。もし、$command を多用する場合には、$command-editor に関する章を読んでいただきたい。

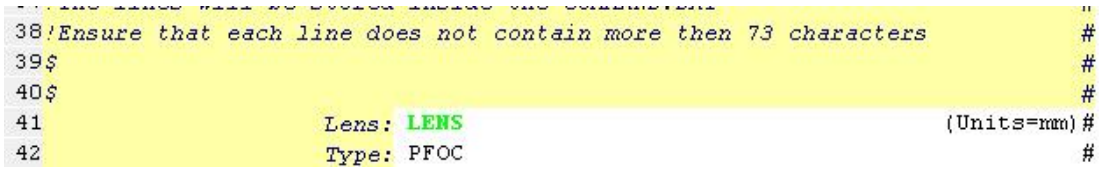
Enter キーを押すと、現在 text caret が存在する command block の$-area に新しい line が挿入される。(註：このとき、$command はパラメーターブロックの中には挿入できない。) 新しい line は”$”で始まり”#”で終わる。ユーザーは、希望の$-command の最初の 1 字か 2 字をこの空の$-line に$のすぐ後ろに入力する。Text caret が他の line に移ると、ScriptWizard は$command を認識し、その行全体を完成する。不明のパラメーターがある場所には疑問符（？）が入る。（もしタイプした$-command が ScriptWizard で理解できない時は、$-line はブランクのまま残る。）このときユーザーは疑問符を意味のある値に入れ替えなければならない。（現在のところ、ScriptWizard は$line の中の数値の有効性をチェックする機能は持っていない。）

COMMAND.DAT ファイルのほとんどすべての line（keyword または subkey を含む line は除く）には line label を付けることができる。これは、10 から 99 までの数字を line の最後にある#の直後に入力すればよい（上記の range の外側のラベルは拒否される）。$-command line は line 全体を選択して BackSpace キーを押せば、削除される。Line を選択する速い方法は、ScriptView エリアの左側で、対応する line number においたマウスをクリックすればよい。

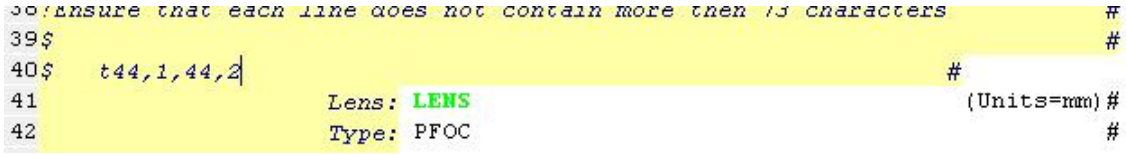
例：
$TAGS#44,ITEM1_44,ITEM2 を挿入するには、次のようにする。まず、text caret をパラメーターエリアのどこかに置く。


!The lines will be stored inside the COMMAND.DAT #
!Ensure that each line does not contain more then 73 characters #
$ #
Lens: LENS (Units=mm)#
Type: PFOC #
LCX,LCY: 0.00000000 0.00000000 #44
NA,FL: 0.85000000 5000.0000 #

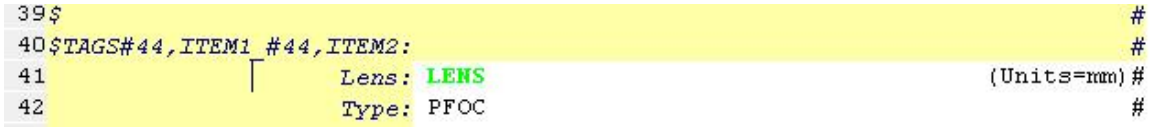
つぎに ENTER を押す。新しい空の$line が挿入される。



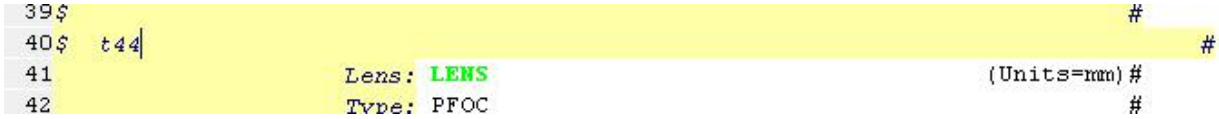
ここで、次のようにタイプする。



Text caret をその line から外すと、ScriptWizard は line を完成する。



ScriptWizard は 1 個またはそれ以上の疑問符(?)が入っている不明のデータを入れ替える。
たとえば、次のようにタイプすると



入力された$TAGS command は次のようになる。

となる。$line を Script から削除するには、line の先頭の”$”を削除すると

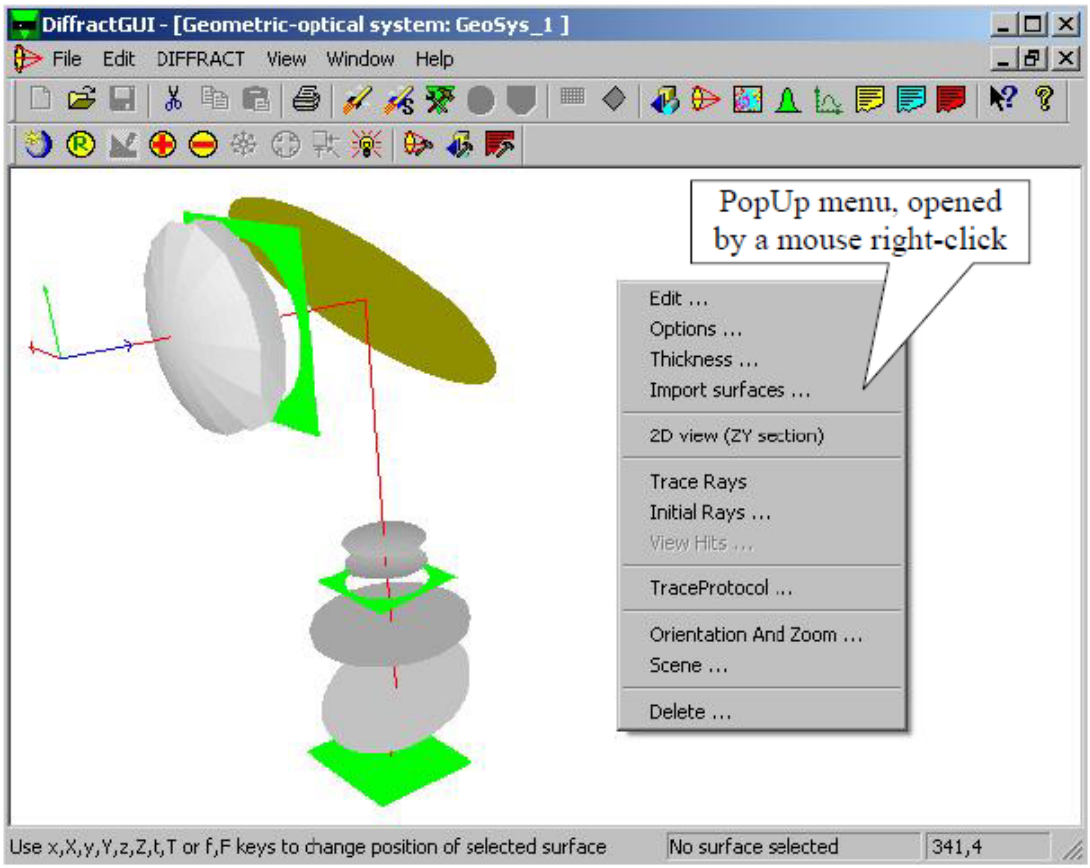
・・・その後 caret を他の line に移す。

・

```
!Ensure that each line does not contain more then 73 characters          #
$                                                                        #
|                  Lens: LENS                                  (Units=mm)#
                   Type: PFOC                                           #
```

7. GeoSysView

この view の内部では-幾何光学システムは 3 次元的に表示できる。COMMAND.DAT ファイル内の GEOP block、または外部の sys-file が **GeoSysView** のソースとして使用される。sys-file は DIFFRACT マニュアルに決められた要求を満たしていなければならない。



幾何光学的面は 3 次元面として表示され、表面の(VCX,VCY,VCZ)座標系で表された位置に置かれる。座標系は 3 色の矢で 3 次元的に表される。青の矢は X 軸、緑の矢は Y 軸、赤の矢は Z 軸である。面の方向は DIFFRACT マニュアルに書いてあるように (θ, ϕ) によって決められる。

もし DiffractGUI に読み込まれた COMMAND.DAT ファイルが GEOP-object を含んでいた場合、ユーザーはそれを 2 ステップで見ることが出来る。まず、**Workbench** 内で GEOP-object を選択し、その後 menu から **View-GeoSys** を選ぶ。DiffractGUI は、選ばれ

た GEOP-object を GeoSysView ウィンドウ内に表示する。すべてのその後の変化は（外部の sys-file から新しいシステムを導入することを含んで）有効となる。しかし、GeoSysView による GEOP-object に対する変更が効果を持つことを確実にするため、変化が終わってから GeoSysView を閉じたほうがよい。

Menu バーから **View-GeoSys** を使って GeoSysView を開き、Workbench に選択された GEOP がない時は、幾何光学システムは表示されない。GeoSysView は空のウインドウとなって残る。ユーザーは GeoSysView の中から（**File-Open** を使って）外部の sys-file を開き、検査と修正を行うことができる。

GeoSysView は Workbench と同じように作用するが、多少の違いと付加機能がある。幾何光学的表面は、常に比例常数、すなわち距離スケールによる DIFFRACT パラメーター値によって決まった位置に描かれる。この比例常数を変更するためには **Edit-ScaleUp** または **Edit-ScaleDown** を使えばよい。（スケールはマウスホイールによって変わる。）この system は全体としてマウスの左ボタンによる drag-and-drop を用いてスクリーン上で回転できる；平行移動は Ctrl を押しながら drag すればよい。

7.1 方向とズーム（Orientation and Zoom）

方向とズームの対話ボックスは、下に示すように Pop-up menu から **Orientation and Zoom** を開けばよい。

通常、画面の方向はスクリーン上で drag すれば変えることができる。（マウスの左ボタンを押しながら、マウスカーソルを動かす。）**X angle** と **Y angle** を決めれば、（**Apply** をクリックするか、Enter を押す）表示された光学面はスクリーン内で回転する。一般には、あらかじめ決められた 6 個の角を直接 **XY, -XY, XZ, -XZ, ZY, -ZY** ボタンに入力すればよい。これらは view point を 3 つの座標軸の上に置く。これらの view point を選択したときに、さらに良い眺めを得るためには、**Ortho** point に切り替えると良い。

CTRL キーを押しながらマウスカーソルで drag したとき、システムはスクリーンの上に変換することができる。しかし、これは **Centred** option を不可能にしたときのみ、作動する。続いて、**Centred** をチェックすれば、前の移動を破棄して、システムはスクリーンの中心に

戻る。

処理を受けた像の大きさは、**Scale** 値で決まる。これは **UP** および **DOWN** ボタンをクリックすることでステップ状に変化する。あるいは、edit box に値を入力（そして **Apply** または *Enter* キーを押せば）するか、またはさらに実際的にはマウスホイールをまわせば、達成できる。

7.2 Scene Option

Pop-menu command Scene を起動すれば、**Scene** 対話ボックスが開く。
左側の図は Scene 対話ボックスの外観を示し、次のような option が可能である。

- **Ortho:** perspective（正射）投影を orthographic（遠近）投影に変更する。これら 2 つのモードの主要な相違は orthographic 投影では、深さは考えていない。

- **Transparency**: チェックをしたときに、幾何光学系の種々の光学面は透明に描かれている。
- **Light:** 描画のコントラストを高くするには、この光源（直接見えない）は点火されている。光源が点火されると表示された光学面の表面のカラー強度は面の法線と面から光源までのベクトルとの角に比例して変化する。
- **Coord .Axes:** スクリーン上の入射座標の 3 軸を隠したり見せたりする（x 軸は赤、 y 軸は緑、 z 軸は青）。
- **Centred**: チェックのとき、ここでは入射座標軸の三脚の原点はウインドウの中心にあり、どのような平行移動も不可能である。

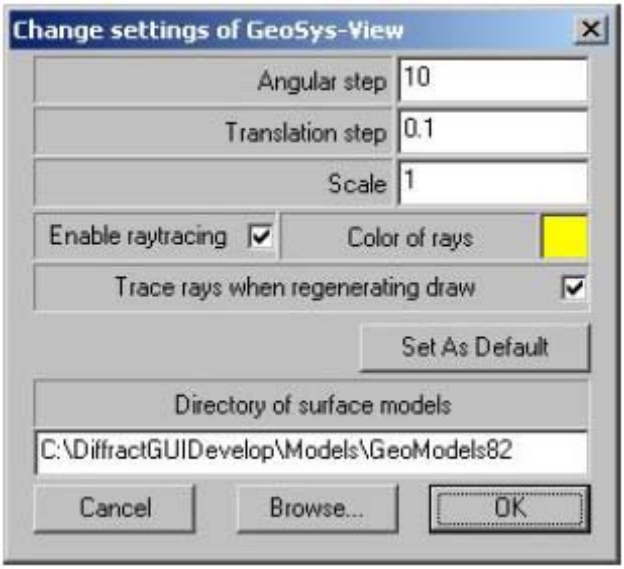
- **Axis size**: ここでは座標軸の長さが決定される。デフォルトでは長さは 1.0mm である。しかし、これはスクリーン上では表示された光学系によっては、短か過ぎたり、あるいは長すぎる。軸の長さを変更するには、適当な値を入力し、Enter を押す。もし、**Ortho** option が可能な場合は、スクリーン上の軸の長さを知ることはスクリーン上の object の大きさを評価するのに有効である。

- **BackColor**: もしチェックを行うと、GeoSysView の背景色 Back Color ボタンの下に

ある正方形に示されている。色正方形をクリックすると、標準の ChooseColor 対話ウインドウが開き、違った背景光を選択することができる。もしチェックをしなかったら、デフォルトの OS 背景カラーが使われる。

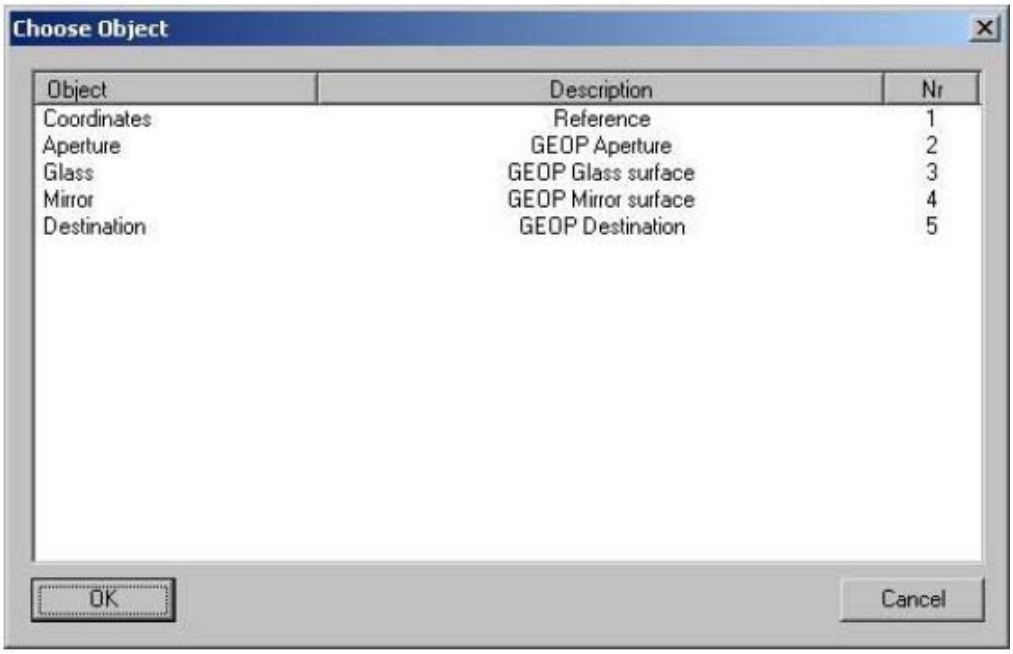
7.3 GeoSysView Options

角度の方向と平行移動のステップは、ユーザーの平行移動または回転の要求で実行される。距離スケールは大きさをスクリーンピクセルで表す。この値は menu の **Edit-Scale Up** あるいは **Edit-Scale Down** で変更し、**Edit-Inflate** または **Edit-Deflate** は変えてはいけない。**Set As Default** ボタンをクリックすると、角度と平行移動ステップは距離スケールと一緒にレジストリーに記録され、次に来る GyoSysView は初期設定としてこの値を使うこととなる。もし、ボタンをクリックしないときは、dialog の中で使われる GeoSysView のみがこれらのパラメーターを使う。種々の光学面の model-file を含む folder の path はここで決めておかなければならない。OK ボタンで閉じる時は、**Set As Default** ボタンをクリックしたかどうかにかかわらず、model path は常にレジストリーに記録される。



7.4 新しい光学面の生成（Creating a new surface）

Geometric-optical_System に新しい光学面を作って挿入するためには、下に述べるように、**ChooseModel** dialog と **SurfacePanel** という 2 つの方法がある。どちらの方法も、希望する光学面のタイプを決め、今選択した面の後ろに新しく作った面を挿入することができる。



もしどの面も選択しなかった場合には、新しい面はリストの最後に追加される。この新しい面は、選択した面とその次の面とを結ぶ直線の中心に置かれ、その方向（θ,ϕ）は先に選択した面と同じ値にセットされる。もともと、面を新設する時には、そのパラメーターはデフォルト値に設定される。新しい面が挿入され、ディスプレイされた後、ユーザーは **Edit DIFFRACT_parameters** dialog を使って必要なパラメーター値を設定されたい。

7.4.1 ChooseModel dialog の使用（Using ChooseModel dialog）

ＣhooseModel 対話ボックスは menu の **Edit-Create** で起動できる。リストされているモデル（どのモデルも DIFFRACT 光学面タイプを表している）の１つを選択し、OK をクリックすると、新しい光学面が作られ、システムに挿入される。

7.4.2 SurfacePanel の使用（Using SurfacePanel）

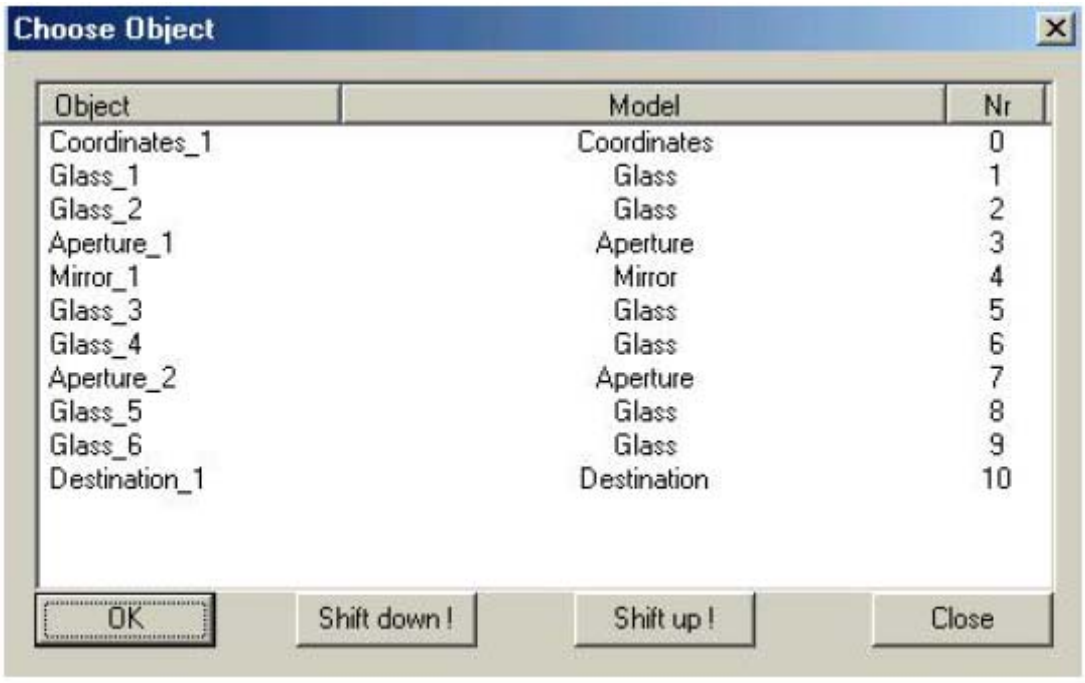
SurfacePanel は、希望する面を表すアイコンをマウスでクリックすることによって、より早く新しい光学面を作る方法である。Menu の **View-SurfacePanel** を選択すれば、起動できる。SurfacePanel の最初の段階でアイコンが描画され、数秒後に SurfacePanel の最終画面となり、**GeoSysView** からアクセスできるようになる。GeoModels folder にある model file の数が、ディスプレイされたアイコンの数を決める。アイコンの形は、model file に記録された情報で決まる。

7.5 光学面の削除（Deleting a surface）

削除する光学面を選択し、キーボードの *Delete* キーを押せばよい。あるいは menu から **Edit-Cut** を選択すればよい。面を選択していないと ChooseObject-dialog が表示され、ユーザーに削除する面を選択する機会を与える。多数の面を選択するときは、Ctrl キーを押しながら、必要な面をクリックして選択リストを作ればよい。

7.6 光学面の順序の変更（Changing the order of surfaces）

光学面の順序の変更は **ChooseObject**-dialog で行うことができる。ChooseObject-dialog を起動する最も簡単な方法は、free background をダブルクリックするか、面を選択しないで menu から **DIFFRACT-Edit Parameters** を選択すればよい。ChoosObject を起動すると、ディスプレイリストから面を選択し、**ShiftUp** または **ShiftDown** キーを使ってリスト内で上下すればよい。



注意：DIFFRACT は面の順序は光線が面に到達する順序に配列していると考えている。DiffractGUI はこの条件のチェックは行わない。

7.7 光学面のパラメーターの変更（Changing the order of surfaces）

7.7.1 Edit DIFFRACT parameters dialog の使用（Using Edit DIFFRACT parameters）

これは Workbench と同じように行われる。**Workbench** の章を見ていただきたい。

7.7.2 キーボードの使用（Using the keyboard）

GeoSysView 内ではキーボードを使って光学面の DIFFRACT パラメーターの変更とスクリーン内での表現を変えることができる。それらは

- **X** または **x** は面の VCX パラメーターを増減する。
- **Y** または **y** は面の VCY パラメーターを増減する。
- **Z** または **z** は面の VCZ パラメーターを増減する。
- **T** または **t** は面の θ を増減する。z
- **F** または **f** は面の ϕ を増減する。

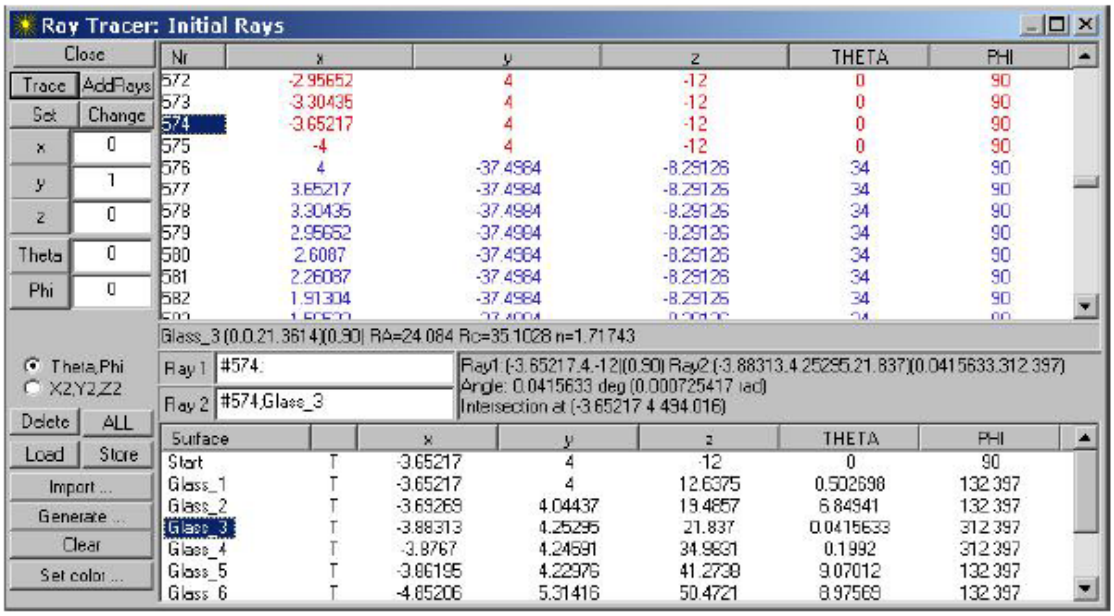
もし Ctrl キーを押したまま x , y, z, t, f キーを押せば、選択した光学面とその後の面全部のパラメーターが変更される。それぞれのキーを押すたびに、値は Option dialog box の **Angular step** および **Translation step** で決めた量だけ増加または減少する。

7.8 光線追跡（Ray tracing）

GeoSysView 内では、DiffractGUI は任意の数の光線について、幾何光学系の光線追跡を行うことができる。光線の座標は幾何光学系の座標軸で表される。それぞれの光線は、destination plane に到達するまで、あるいは考えている面の開口の外側に来るまで続けられる。すべての光線はスクリーン内に（**Geosys Options**-dialog を使って決めた色で）表示される。光線追跡は **Enable ray-tracing** option を起動したときのみに行われる。**Trace Rays When Regenerating Draw** を起動すると、DiffractGUI は **Regenerating Drawing** を選ぶ度に光線追跡を行う。多くの光線を選び長い計算時間が掛かる時、この option は起動しない方がよい。このようなときには、pop-up menu の **Trace rays** を使って光線追跡を行う。

7.8.1 Initial Ray の変更（Changing the Initial Rays）

Pop-up menu から **Initial Rays** dialog を起動すると、initial ray のセットを変更できる。デフォルト設定では、この dialog を最初に開いた時、座標 $(\theta, \phi, x, y, z) = (0,0,0,0,0)$ の 1 本の光線のみが存在する。

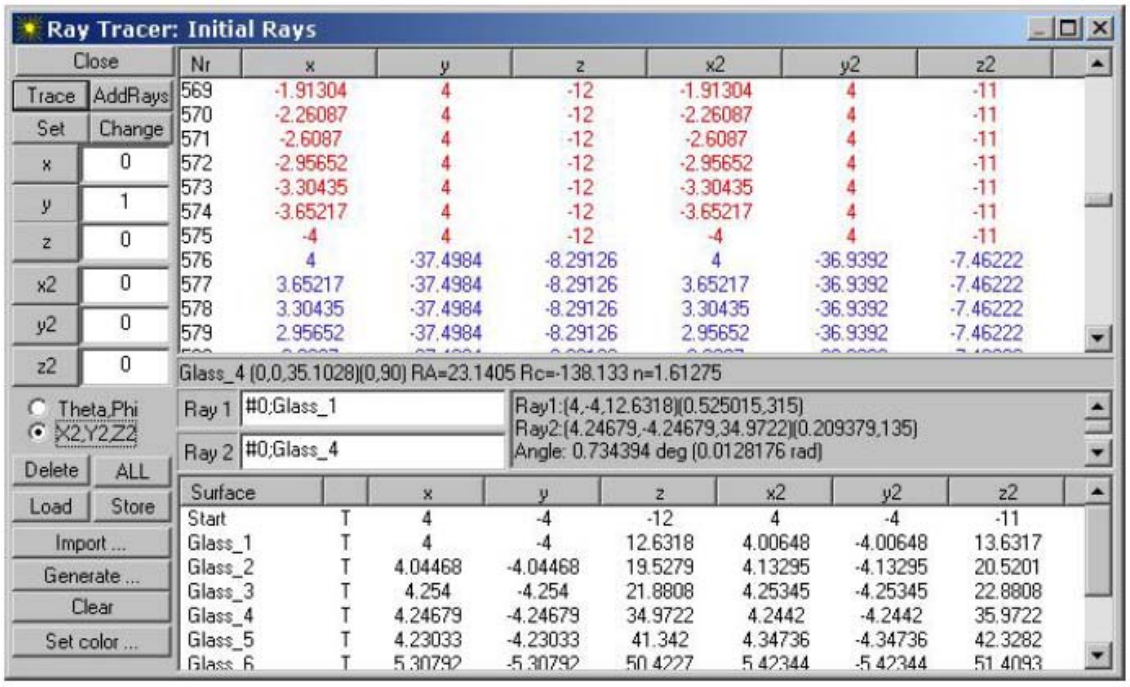


上のリストから、1 本またはそれ以上の光線を選択できる。*Delete* キーは選択した光線（もしあれば）だけを削除できる。もし全部の光線を削除したい時は、すべての光線を選択した後で、*Delete* キーを押せばよい。$dx, dy, dz, d\theta, d\phi$ を設定して新しい光線をリストに追加することができる。追加された光線は、選択された光線の座標に対し、それぞれ $dx, dy, dz, d\theta, d\phi$ がプラスされる。

もし、光線を選択しないときには、新しい光線がリストに加えられる。1 本またはそれ以上の光線が選択されると、その選択された光線の次に新しい光線が挿入される。最後に、**Store** および **Load** ボタンにより、リストが保存され、再び読み出される。光線リストは（人間の読める）ASCII ファイルとして保存されている。

この dialog box は **GeoSysView** session の間開かれている。このことはユーザーが session を閉じることなくいつでも **GeoSysView** に戻り、パラメーターの変更ができ、**InitialRay** に切り替えることができる。しかし、（たとえば **LoggerView** に切り替えて）GeoSysView が閉じたとき、この dialog box は自動的に閉じられる。

初期光線は射出点と 2 つの角 (θ, ϕ)、または空間の第 2 の点の座標 $(x2, y2, z2)$ で特定される。2 個のラジオボタン **Theta, Phi** および **X2, Y2, Z2** はこれらの表示を切り替えることができる。



Theta および dPhi の edit box は dx2 および dy2 に置き換えられ、新しい dz2 という edit box が **X2, Y2, Z2** モードで現れる。これらは **Theta, Phi** モードにおける dTheta および dPhi box と同様に使うことができる。

上のリストで 1 本の初期光線を選択し、光線追跡を実行すると、その光線と光路中の種々の面との交点が対話ボックスの下のウインドウに表示される。このリストは編集できないが、マウスの左および右クリックによって 2 本の光線を選ぶことができる。選択した光線の名前は 2 個の窓 **Ray1**, **Ray2** に表示される。両方の光線が決められる（左クリックで **Ray1**, 右クリックで **Ray2**）と、DiffractGUI はそれらの間の角度と、（もし交わるなら）両者の交点の位置を計算する。その結果は dialog box の右下部の **Ray1** と **Ray2** の下にある小さな窓に表示される。

光線リストの中の各光線にはそれぞれ独特の色を付けることが出来る。**Set color…**ボタンをクリックすると ChooseColor ボックスを開くことができ、ユーザーは任意の色を決めることが出来る。この色は、前に選択した光線に適用することができる（または **Set Color…** ボタンをクリックする前に特定の光線を選んでいないときは、すべての光線に適用することになる）。

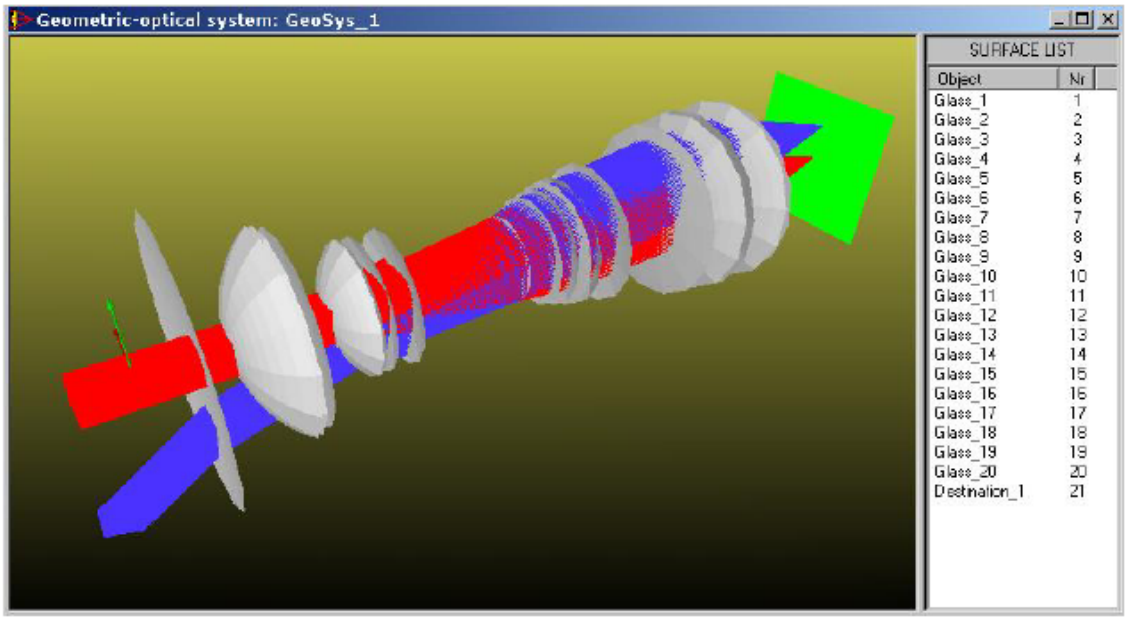
Clear ボタンはすべての複製した光線と、幾何光学系の最初の面に入射しない光線を取り消すことが出来る。（**註**：これらの光線は前もってシステム全体を追跡しておかなければなら

ない。そうでなければ **Clear** ボタンはリストにある全部の光線を消去することになる。）

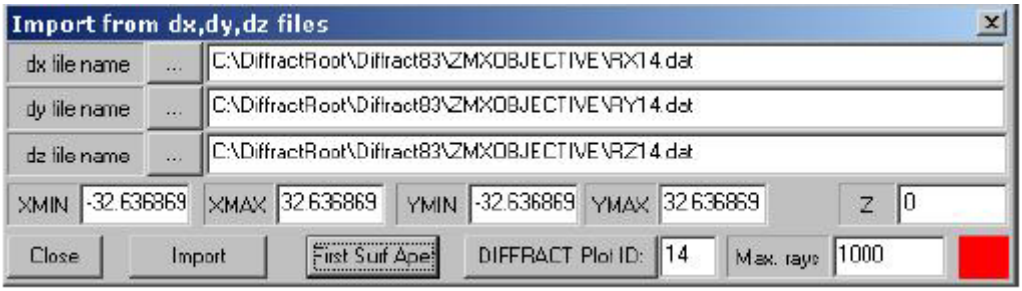
Generate…および **Import…**は光線の多数の扇状光線束を作るとか、(DIFFRACT によって作られ光線描画ファイルのような）外部ファイルから入射光線を導入するのに役立つ。さらに詳しいことはこの章で後に述べる。

Delete は選択した光線をすべて削除する（もし、光線を選択しないと作動しない)。（これまで選択したかどうかに関わらず）すべての光線リストを削除するときは、Delete ボタンの右にある All ボタンをクリックすればよい。

Set および **Change** ボタンはリスト中の選択した光線について、特定のパラメーターを変更するときに用いられる。ここでも、どの光線も選択していないと、すべての光線が変化を受ける。これらのボタンを使う前には、**x, y, z, Theta, Phi**（あるいは X2, Y2, Z2 における **x, y, z, x2, y2, z2**）のパラメーターをチェックして、どのパラメーターを変更するかを決めてから、対応する訂正画面に入力しなければならない。**Set** は目的とする光線に選んだ値を設定するが、**Change** は選択した光線の現在の値に選んだ値を加算する。次の図は、赤および青に着色した 2 組の光線による光線追跡の例を示す。



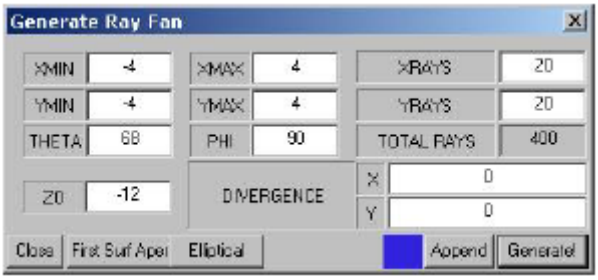
…**Import..**によって起動される import-ray dialog box によって、方向ベクトルが dx, dy, dz で表されている 1 組の光線を導入することが出来る。



ファイル名は任意に選ぶことが出来るが、この dialog box は特に DIFFRACT の PLOT 命令で作られた 3 つのファイル（RXnn, TYnn, RZnn）によって導入される光線に対して使われている。通常、この描画は command.dat ファイルの conversion-to-rays 命令によって直ちに実行される。**DIFFRACT Plot ID**:ボタンは入力された ID ナンバーによってファイル名を DIFFRACT の作業ディレクトリ内に自動的に作る。完全を期するために、“**...**”ボタンは他のファイルを表示するために用いられる。

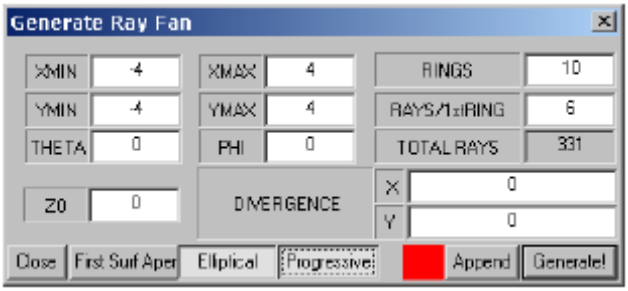
光線はＸＹ面の中の矩形 ***XMIN, XMAX, YMIN, YMAX*** の中のＺの位置に一様に分布している。**First Surf Aper** ボタンはこれらの分布がこのシステムの第 1 面の開口をカバーするように作用する。

光線が満杯になるのを防ぐため、入射する光線の数を **Max.rays** で決められた値に制限することが出来る。入射する光線の数は光線のスキップが起こるため異なることがある。スキップされた光線の数を増大することは、制限と矛盾することではないことを保障しているが、入射した光線の数は認識される光線の数と正確に一致することを保障することは出来ない。光線の色は dialog box の下右側の色ボタンをクリックして確定することが出来る。一旦すべての希望する光線が確定したときは、**Import** ボタンで入射を初期化することが出来る。



Initial_Ray dialog box における **Generate...**をクリックして開くことが出来る Ray-generator dialog box は与えられた位置と方向を持つ平面から射出する光線を定義することが出来る。***XMIN, XMAX, YMIN, YMAX, THETA, PHI*** および ***Z0*** は射出面を決める。こ

れらの光線は矩形平面内に矩形状または同心円状に分布する。上の図では 20 行 20 列の 矩形状にセットされている。もし DIVERGENCE X および Y の値（度であらわす）が 0 にセットされれば光線は互いに平行であるが、正（負）の値は拡散する（収斂する）扇状光線束を表す。 x および y 方向の拡散角は独立に決めることが出来る。



光線の円形分布（Circular distribution）は、上に示すように、**Elliptical** オプションをチェックすると実行することが出来る。この場合、**Progressive** オプションが可能である。もしこれをチェックすると、与えられたリングの光線の和は、次の（内部の）リングの光線の数の 2 倍となるが、そうではないと各リングの光線数はすべて同じとなる。

Append オプションは、新しい扇状光線束を作る前に Initial Ray dialog（光線の集まりを制御する）を消去するのを防止する。したがって、このオプションによって、多数の光線束を容易に作り出すことが出来る。光線の色は **Append** ボタンの左側にある色の付いた正方形をクリックして選ぶことが出来る。

First Surf Aper ボタンはシステムの第 1 面をカバーする ***XMIN, XMAX, YMIN, YMAX*** の値を作ることが出来る。XMIN の数を入力しながら、XMAX, YMIN, YMAX は連続的に更新される。したがって、矩形ではない値を入力するときには、最初に XMIN を入力しなければならない。そうしないと、最初に選んだ XMAX, YMIN, YMAX の値がが無効となる。

最後に、**Generate!**ボタンによって光線束が形成され、GeoSysView の Initial_Rays リストに従って配置される。

7.8.2　厚さの dialog box (Thickness Dialog Box)

***Thickness of a surface*（光学面の厚さ）**に関して言えば、*DiffractGUI* ではその面の頂点からすぐ次の面の頂点までの距離を意味する。Pop-up menu command **Thickness** によって起動するこの小さな対話ボックスの中で、ユーザーはこの距離をチェックし、x, y, z 方向のずれを決めて厚さを変更することができる。変更するためには、ずれの値 dx, dy and/or dz を変更し、**Apply changes** を選べばよい。この対話ボックスのタイトルとして表示している表面（上の例では Glass_8）の厚さのみが変化を受け、他の面の厚さは変わらない。したがって、後続するすべての面は、それらの厚さを変えないように、全体として移動することに注意されたい。この対話ボックスは開いたまま（モードレス：表示されているがクリックしない限り動作しない。）残され、新しい面が選択される毎に自動的に内容が更新される。

7.8.3 光学面との交差（Hits on Surfaces）

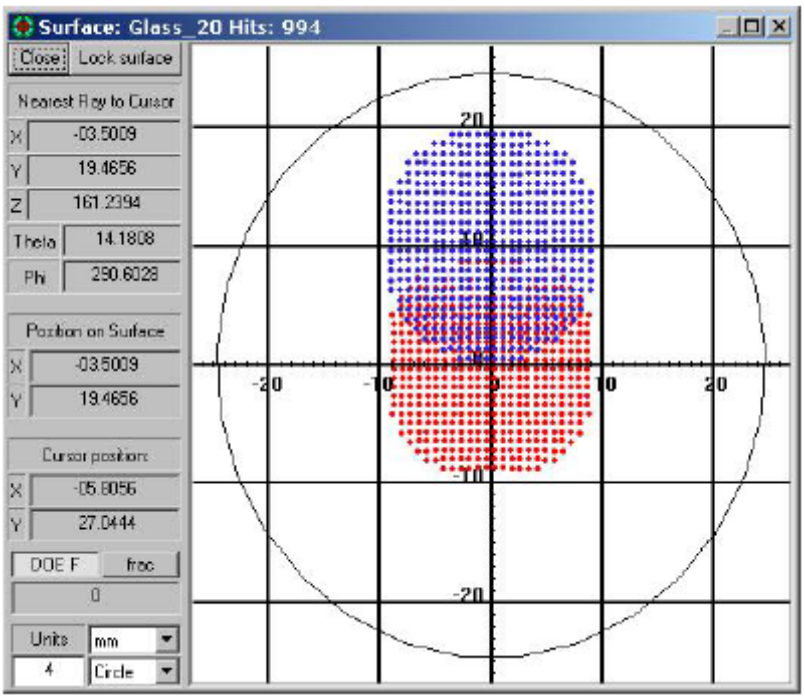
それぞれの光学面において、与えられた開口の中で交差した光線は、Hits dialog で見ることが出来る。これは pop-up menu から **View Hits on** を選べばよい。

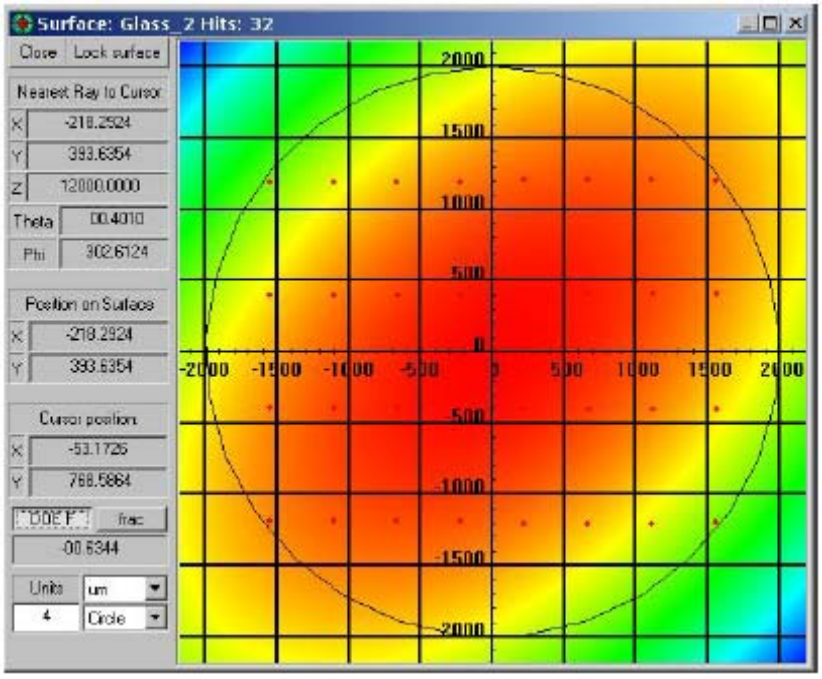
マウス・ホイールは Ray-Hits dialog box のグラフを拡大・縮小に使うことが出来る。中央のマウスボタン（通常ホイール）を押し込むとグラフが左右に拡大される。左ボタンをダブルクリックすると、元の広がりに戻る。右ボタンをダブルクリックすると矩形（すなわち Xmin=Ymin, Xmax=Ymax）となり、すべての光線をカバーする。これは、特に最終面に到達するところを表示するのに役に立つが、それは最終面はグラフの初期広がりを決める開口を持たないからである。

光線が到達する点の形と大きさは dialog box の左下部にある部分によって決められるからである。

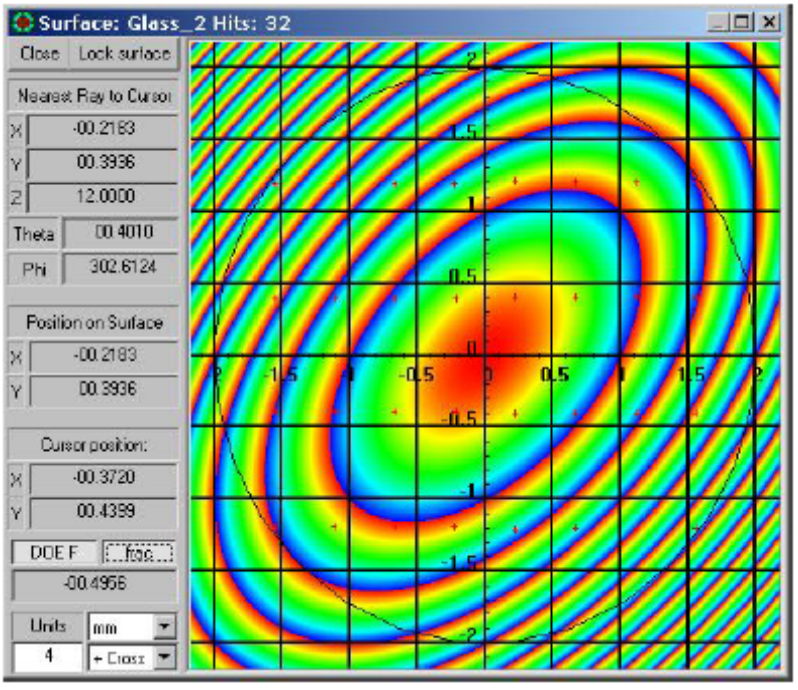
初期設定では、dialog box は自動的にシステムの選択した光学面における到達点を表示する。**Lock Surface** チェックボタンを押すとこの表示方法が無効となる。したがって、現在選択している面の如何にかかわらず、**Lock Surface** がチェックされたときに選んだ面における

到達点が表示される。さらに、このモードではグラフの限界はそれぞれの更新時点でリセットされず、光線追跡の度に re-zoom または re-pan の必要なしに開口の sub-area での結果を見ることが出来る。





次の図における円は光学面の開口を表す。すべての位置の単位は combo box に示されているが、この場合は mm、角度は度である。面のグラフの内部に Zooming 正方形を認識することができ、これはマウスボタンによる drag-and-drop で広げることが出来る。もしグラフが既に zooming 済みの場合は、正常サイズに戻すことができるる。

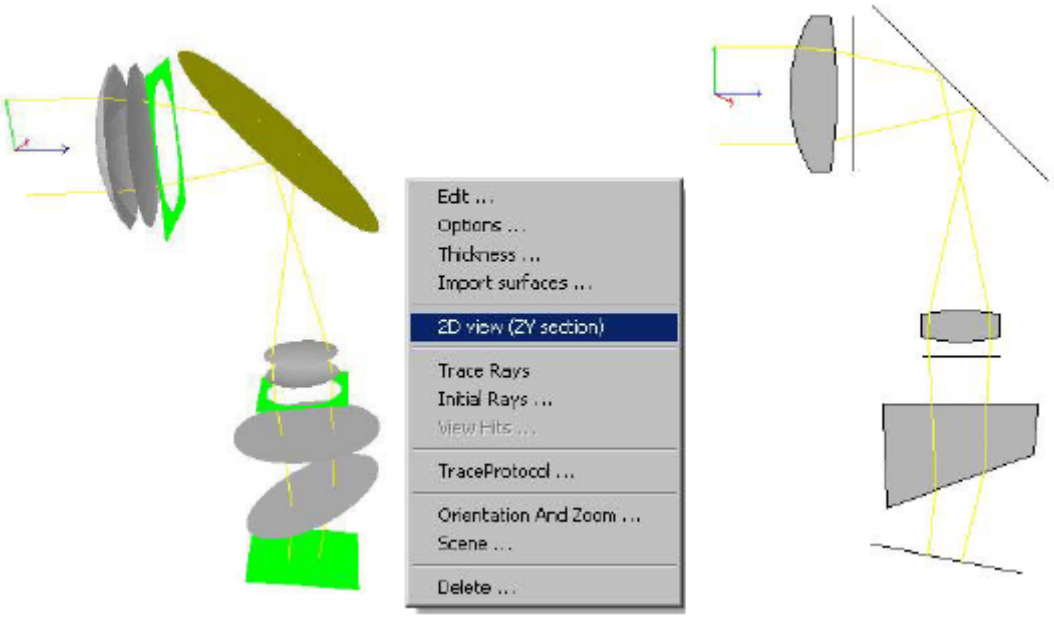


もし光学面が DOE コートされているときは、***DOE F*** option を起動すると、DOE の多項式関数 F®,または F(x,y)が（図に見られるように）カラーbitmap として背景に表示される。カーソルの位置における関数の数値はカーソルの動きと連動して表示される。***frac*** optio を起動すれば、上図のように DOE コーティングの厚さの変動部分のみが表示される。

この対話ボックスはモードレスタイプである。これは、対話ボックスを閉じなくても **GeoSysView** に戻ることができる。もし、幾何光学システムで他の光学面を選択する（たとえば TAB キーを押して）と、対話ウインドウの内容は活性化され、その面に入射する光線が指示される。

7.9 幾何光学系の ZY 断面の表示（View the ZY-section of a geometric-optical system）

pop-up menu command **2D View** を起動すると、**GeoSysView** は幾何光学系の ZY 断面を表示する。



この表示では、システムは回転できる。座標軸は常に上の図に示されるように向いている（右側の図を参照されたい）。 z 軸は左から右に向き、 y 軸は下から上に向いている。**2DView** は光学面の YZ 面による断面が描かれている。もし与えられた光学面について、その中心座標が YZ 面上にないか、あるいは面の法線が YZ 面にないときは、その面は 3 次元 objectt として表示される。

2Dview の他の機能は、2 つの隣接する面の間の空間が 1.0 ではない屈折率をもつときには

グレイに影付けされる。（自由空間ではない媒質が存在することを示している。）

GeoSysView の種々の特性、たとえば光線追跡、システムパラメーターの変更などは、**2DView** でも有効である。

7.10　外部ファイルによる GEOP 光学面の導入(Importing GEOP Suirfaces from External Files)

DiffractGUI における **GeoSysView** では、ユーザーは外部ファイルによって 1 個またはそれ以上の光学面を導入することができる。この特性を使うためには、右マウスクリックにより pop-up menu を開き、***Import surfaces*** を選ぶ。（導入した面は、現在選択した光学面の直後に挿入される。どの面も選択しない場合には、現存光学面リストの最後に置かれる。）新しく導入された光学面は、外部ファイルの対応列で決められたパラメーター値を想定している。

7.10.1　ファイルの文法（File syntax）

外部ファイルは数行のテキストをもつテキストファイルで無ければならない。添付した *ZoneLens.TXT* は典型的な例である。各行は空白あるいは Tab で区分した、多くのエレメント（実数値）を含んでいる。テキストの第 1 行（すなわち第 1 行、または Table Header）は次のようにコラムを規定している。

- Table Header の最初のエレメントは 4 文字の言葉 *ZONE*、あるいは 5 文字の言葉 *BLOCK* で無ければならない。*DiffractGUI* は行の第 1 エレメントとしての *ZONE* あるいは *BLOCK* というキーワードまではすべて無視する。次いで、この行は次の *ZONE* あるいは *BLOCK* が見つかるまでは Table Header とみなす。*ZONE* あるいは *BLOCK* と次の *ZONE* あるいは *BLOCK* の間の行数は制限が無い。

- 第 2 コラムから並ぶエレメントは Table Header におくべきエレメント、すなわち *Glass, Mirror, Aperture* および *Destination Plane* などの光学面に用いるパラメーターの名前でなければならない。たとえば非球面ガラス面の場合には、これらは頂点座標 *VCX, VCY, VCZ,* 頂点における表面法線の方向角、θ，φ，曲率半径 *RC,* 円錐係数 *K,* 非球面係数 *A1* から *A21* まで、開口半径 *Rmin, Rmax,* 屈折率 *NREFRACT,* などであり、これらが Header におかれるエレメントである。

Table Header に続く行には、*DiffractGUI* は *GEOP* システムにガラス面（*Glass*）、ミラー

面(*Mirror*)、開口絞り(*Aperture*)、最終面（*Destination Plane*）を挿入する。各行には少なくとも Table Header と同じ数のエレメントを置かなければならない。そうでないと、その行で定義された光学面は挿入できない。（もしある行が Table Header より多い数のエレメントを含んでいるときは、余分のエレメントは無視される。）

Table の第 1 コラム（すなわち、Header *ZONE* または *BROCK* の下）には、任意のテキストを置いてよいが、*DiffractGUI* は次のような最初の文字だけを認識する：*G* は Glass Surface, *M* は Mirror Surface, *A* は Aperture, *D* は Destination Plane と理解する。もし、与えられた行の最初のエレメントの第 1 文字が"！"とすると、*DiffractGUI* はその行をコメント行とみなし、無視する。

DiffractGUI の configuration（構成）が DIFFRACT 7.7 あるいは 7.9 と組み合わせて使うときは、*ZONE* および *BLOCK* Keyword は同じ意味に用いられている。しかし、DIFFRACT 8.1 またはそれ以後の Version と組み合わせるときは、*ZONE* グループの光学面は *BLOCK* グループの光学面とは違っていて、これらは Hit and Miss パラメーターと関連付けて使われている。*BLOCK* の後にリストされた光学面は Miss = 0 で、Hit パラメーターは自動的に次の光学面に向かっている。*ZONE* の後にリストされた光学面は次の面に向かって Miss となり、その *ZONE* グループが終わった後の最初の光学面に向かって Hit となっている。(Hit および Miss ははっきりと割り当てられた値を持っていない唯一の表面パラメーターであることに注意されたい：これらの値は導入されたファイル内でグループ化と配置順序によって暗黙的に決められている。)

Table Header にリストされたパラメーター名は、その順序に並んだ後続行のコラムの値を *DiffractGUI* が読み取り、それを後続行で順次指定する新しい光学面に割り当てる役割を果たしている。次のようなパラメーター名が使われている。

SSHAPE と名づけられたコラムは、*DiffractGUI* にどのような種類の光学面を作るかを決めさせる。そのようなコラムは数値は含んでいないが、次のような keyword の中の一つを含んでいる。**Flat, Spherical, Conical, Aspherical, Cylindrical,** あるいは **Axiconical** である。Keyword 認識は大文字・小文字を区別し、Keyword は略してはいけない。

VCX, VCY, VCZ, THETA, PHI は各面の位置と方向を決めている。また、*DiffractGUI* は外部ファイルで導入された第 1 面の値を基礎にして、それぞれの新しい光学面の位置と方向を計算するために、*DeltaZ* あるいは *Thickness* というパラメーター（下を見よ）を使うことができる。しかし、導入ファイルで *VCX, VCY, VCZ, THETA, PHI* と名づけられたコラムを挿入すると、*DeltaZ* あるいは *Thickness* による決定は無視されることになる。（す

なわち、決められた *DeltaZ* あるいは *Thickness* 値に関係なく光学面の位置が決まる。）
DeltaZ は考えている面の頂点とこの面を含む *ZONE* あるいは *BLOCK* グループの第 1 面からの Z 軸方向の距離である。*DeltaZ* は DIFFRACT の表面パラメーターではないが、これを定義しておくと、*DiffractGUI* は表面の *VCX, VCY, VCZ, THETA* および *PHI* パラメーターを *DeltaZ* の値から計算することができる。新しい *ZONE* または *BLOCK* グループの第 1 面に対し、*DeltaZ* は前の *ZONE* または *BLOCK* グループの第 1（先頭）面までの距離を決めている。もしそのようなグループが存在しない場合は、*DeltaZ* は座標軸の原点から Z 軸に沿った距離と考えられる。

Thickness は考えている表面の頂点から次の表面の頂点までの距離を規定する。*DeltaZ* と同様に、*Thickness* は DIFFRACT における表面パラメーターではないが、これを定義することによって、*DiffractGUI* は現在の表面の *Thickness* によって、次の表面の *VCX, VCY, VCZ, THETA*,および *PHI* を算出する。

Rc, **K** は、ガラス面および鏡の曲率と円錐係数を決める。

A1, A2, A3,…,A21 はガラス面および鏡の非球面係数である。
円筒面のときは､円筒軸の PHI と LA パラメーターは **LPHI** および **LLA** というコラムで規定されるべきものである。
アキシコン面の場合、アキシコンの半頂角は **AXIANGLE** のコラムに規定されるべきである。

MEDIUM は次の keyword の一つを含むコラムの名前である：**Free, Homo** または **Grin**。*SSHAPE* の場合と同じように、ユーザーは keyword が大文字・小文字に依存し、略記をしてはいけない。

NREFRACT はガラス面に次ぐ均質媒質の屈折率である。

NI, KI は鏡の複素屈折率(n,k)を決める。

GRIN 媒質に対して付け加わった ***OX, OY, OZ, GTHETA, GPHI, N0, R0, Z0, Q1*** から ***Q6***, ***W1*** から ***W12*** を決めなければならない。GRIN の形式は **GrinType** で設定できるので、パラメーターは 1, 2 のみである。

*コーティングは keyword TCOAT（ガラス面）または RCOAT（鏡）で規定される。可能な option は **AR, DOE, BARE, No**（BARE と同じ）または **Yes**（AR と同じ）である。DOE*

*コーティングパラメーターは **DOEORDER, DOELAMBDA, DOEXPHI** であり、平面ではない DOE 係数は **B01, B02, B03** から **B21** まで、平面の DOE 係数は **B1x0, B0x1, B2x0, B1x1, B0x2, B3x0, B2x1, B1x2, B0x3, B4x0, B3x1, B2x2, B1x3, B0x4, B5x0, B4x1, B3x2, B2x3, B1x4, B0x5, B3x3, B6x0, B5x1, B4x2, B3x3, B2x4,B1x5** および **B0x6**. である。*

Rmin, ***Rmax*** は、開口の半径の上限および下限である。*Rmax* コラムは常に存在するが、*Rmin* はその値が 0 のときは省略しても良い。(DIFFRACT 7.7 および 7.9 では、*Rmin* 値)は明白なら無視しても良い。

導入された *ZONE* あるいは *BLOCK* の各面のパラメーターは、Table のの中で ***DEFAULT*** という keyword で始まる行で ***Name=Value*** で決める任意の値を設定してもよい。たとえば、*ZONE* あるいは *BLOCK* におけるすべてのガラス面は同じ屈折率を持っているとすれば、次のような行を屈折率に対して設定する。

DEFAULT　NREFRACT = 1.567

この option によって、すべての行に同じ値を書くコラム（この例では *NREFRACT*）を加える必要をなくすることができる。この割り当て宣言は空白や Tab 文字を含んではいけない。(たとえば　*DEFAULT NREFRACT= 1.567* は、= 符号の後に空白がある。）空白や Tab はある設定を他の設定から区別するときに用いられる。

註：

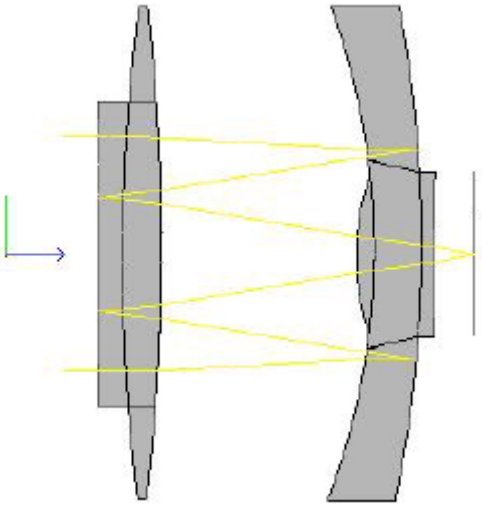
- すべての設定は **mm** であると考えている。
- 新しい光学面についてすべてのパラメーター値を設定する必要は無い。*DiffractGUI* は **SurfacePanel** によって新しい光学面を挿入したときと同じ default 値を採用する。
- もしコラムのタイトルが未知のパラメーター名であったときは、そのコラムにある値は無視される。これによって異なったタイプの光学面のグループが挿入されるのを防いでいる。
- 挿入ファイルは *Microsoft* Excel のようなスプレッドシートプログラムで作ればよい。（スプレッドシートを保存するときは、"*Save as *.txt*"という option を使うこと。）

7.10.2　例

1. 次の表は **BLOCK** と T**HICKNESS** の使い方を示す。

```
! All surfaces have spherical shape
DEFAULT SSHAPE=Spherical
! ============================================================
! This surface will be placed always at the position (0,0,20)
BLOCK Rc       Thickness NREFRACT Rmax THETA PHI VCX VCY VCZ
G1    340.785  6.5       1.517    42   0     0   0   0   20
! ============================================================
BLOCK Rc       Thickness NREFRACT Rmax THETA PHI
G2    -375.235  36        1         42   0     0
G3    -120.616  8         1.517     42   0     0
! ============================================================
BLOCK Rc       Thickness Rmax THETA PHI  NI     KI
M4    -215.82  -8        42   0      0   0      1
! ============================================================
BLOCK Rc       Thickness NREFRACT Rmax
DEFAULT  THETA=180   PHI=90
G5    120.616  36        1         42
G6    375.235  6.5       1.517     26
G7   -340.785  4         1.517     26
! ============================================================
BLOCK Rc       Thickness Rmax THETA PHI NI KI
M8    1316.482 -4        26   180   90  0  1
! ============================================================
BLOCK  Rc       Thickness NREFRACT Rmax
DEFAULT THETA=0 PHI=0
G9     340.785 6.5      1.517     26
G10   -375.235 33       1          26
G11    41.443  3        1.517     16
G12   -120.616 8        1.517     16
G13   -215.82  2        1.728     14
G14    379.752 6.783    1         14
! ============================================================
! the destination plane. Since the 'Thickness' of G14 is 6.783,
! it will be placed 6.783 mm beyond G14
BLOCK
D15
! ============================================================
```

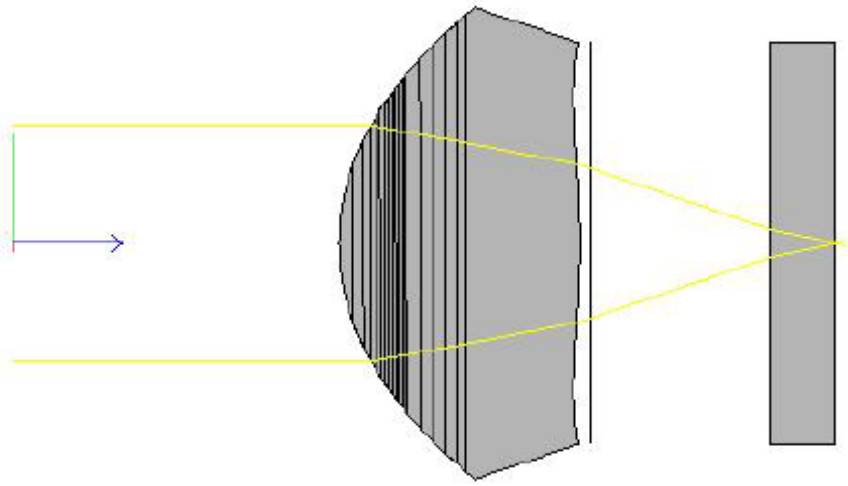
上記の表は次の幾何光学系（2 次元表示）を生成する。



2. 次の表は ZONE と DeltaZ の使い方を示し、下図にその幾何光学系を示す。

```
! Set refractive index of the zoned lens to 1.54056
DEFAULT NREFRACT=1.54056
! ============================================================
! The first zone will be placed at (0,0,3), because DeltaZ=3
ZONE Rmin  Rmax   DeltaZ       Rc       K          A4
0    0.000 0.733  3.00000E+00 2.10100 -0.50000 3.92000E-04
1    0.733 0.957 -1.33722E-03 2.10147 -0.50000 3.91700E-04
2    0.957 1.081 -2.67444E-03 2.10194 -0.50000 3.91400E-04
3    1.081 1.171 -4.01166E-03 2.10240 -0.50000 3.91100E-04
4    1.171 1.243 -5.34888E-03 2.10287 -0.50000 3.90800E-04
5    1.243 1.303 -6.68611E-03 2.10334 -0.50000 3.90500E-04
6    1.303 1.355 -8.02333E-03 2.10381 -0.50000 3.90200E-04
7    1.355 1.401 -9.36055E-03 2.10428 -0.50000 3.89900E-04
8    1.401 1.442 -1.06978E-02 2.10474 -0.50000 3.89600E-04
9    1.442 1.480 -1.20350E-02 2.10521 -0.50000 3.89300E-04
10   1.480 1.514 -1.33722E-02 2.10568 -0.50000 3.89000E-04
11   1.514 1.546 -1.47094E-02 2.10615 -0.50000 3.88700E-04
12   1.546 1.576 -1.60467E-02 2.10662 -0.50000 3.88400E-04
13   1.576 1.722 -1.58003E-02 2.10654 -0.80000 4.11592E-03
14   1.722 1.851 -2.18773E-02 2.10868 -0.80000 4.10477E-03
15   1.851 1.956 -2.79543E-02 2.11081 -0.80000 4.09362E-03
16   1.956 2.045 -3.40314E-02 2.11294 -0.80000 4.08247E-03
17   2.045 2.122 -4.01084E-02 2.11507 -0.80000 4.07131E-03
18   2.122 2.191 -4.61854E-02 2.11721 -0.80000 4.06016E-03
!============================================================
! G19 will be placed 2.21 mm after zone 0
! G20 will be placed 1.74 mm after G19
BLOCK Rmax   DeltaZ  Thickness   Rc      K   A4            MEDIUM
DEFAULT NREFRACT=1
G19   1.866  2.21    1.74       -8.45    0  -0.016020000  Free
! ============================================================
! Note that NREFRACT as table header overrides the default
BLOCK SSHAPE Rmax   Thickness  MEDIUM NREFRACT
G20   Flat   1.866   0.6       Homo   1.5798200
G21   Flat   1.866  -2.25      Free   1
! ============================================================
BLOCK
Destination
```

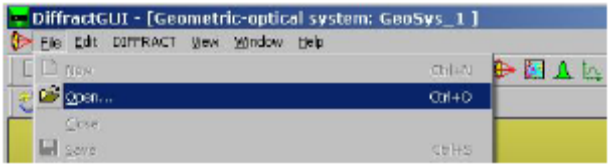
上記の表は次の幾何光学系（2 次元表示）を生成する。



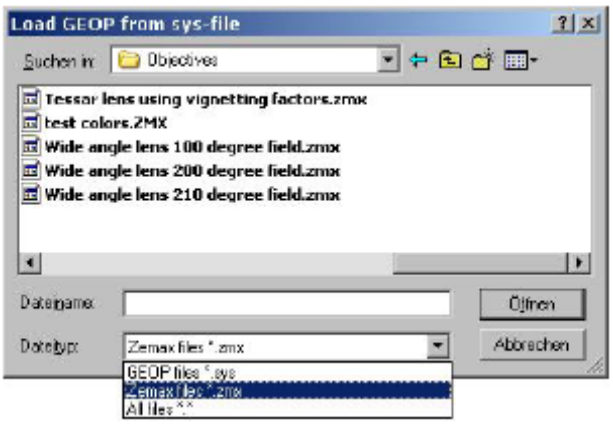
7.10.3　Zemax（*zmx）ファイルの導入(Import Zemax(*.zmx) files)

GeoSysView は Zemax™ ファイルを DIFFRACT の GEOP オプションに導入することが出来る。

ユーザーは GeoSysView を開き、メニューから File-Open コマンドを選択する。



File-Open dialog box では”Zemax files *.zmax”タイプを選び、希望するファイル名を選んで Open をクリックする（または ENTER を押す）。

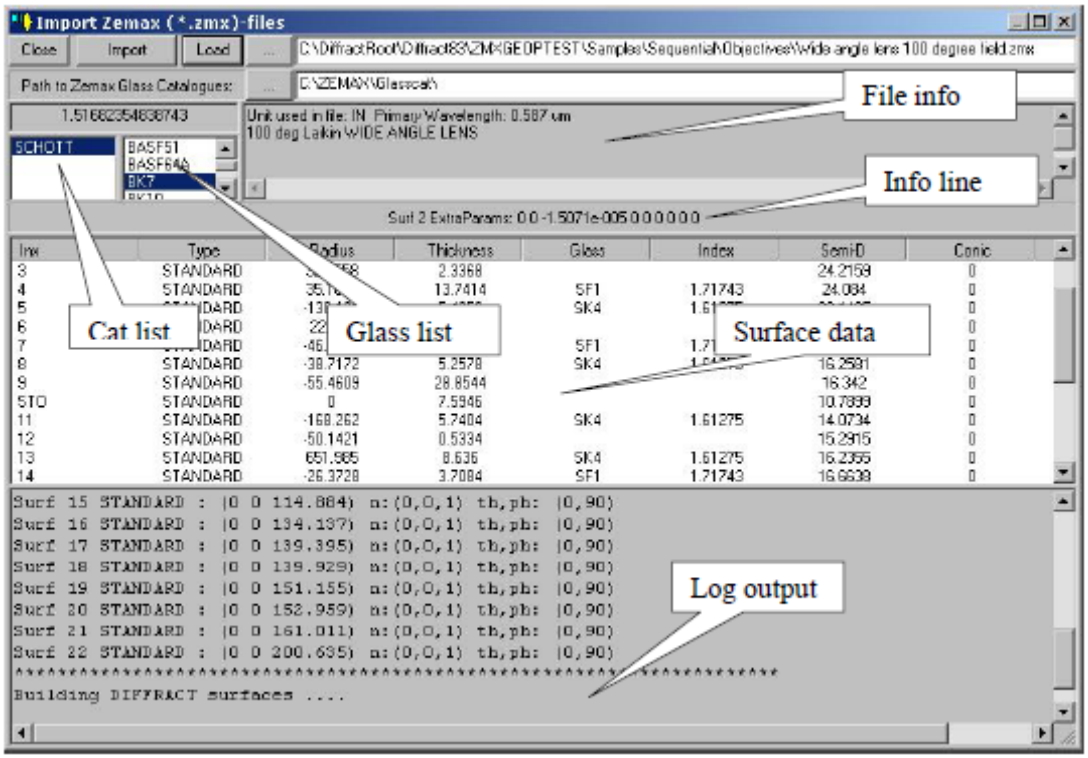


こうすると次の図に示すような中間的な Zemax_Import dialog box が現れる。 ZemaxTM フィアルを導入するためには、2 段階のプロセスが必要である。

1. LOADING： 最初に zmx ファイルを Zemax_Import dialog box に導入する。これによってその内容が表示され、現在使用中の GOEP を破壊することなく、導入できない光学面のタイプを認識させ、未知のガラス材料その他の詳細を示す。この第 1 ステップは Load ボタンによって実行される。

2. IMPORTING： 導入された Zemax™ 光学面を DIFFRACT™ 光学面に転換するために、Import ボタンを押す。その結果として、現在の幾何光学システム(GEOP)に含まれる DIFFRACT™ 面は破棄され、GEOP は新しい光学面で構成される。

正しい load と import では DiffractGUI は zmx ファイルに使われているのと同じガラスカタログを利用しなければならない。このため、ガラスカタログファイル（典型的にはたとえば “C: ¥Zemax¥ Glasscat¥”）を決めなければならない。このフォルダーの中には、*.agf ファイルのみが loading には必要である（*.bgf ファイルは考えていない)。



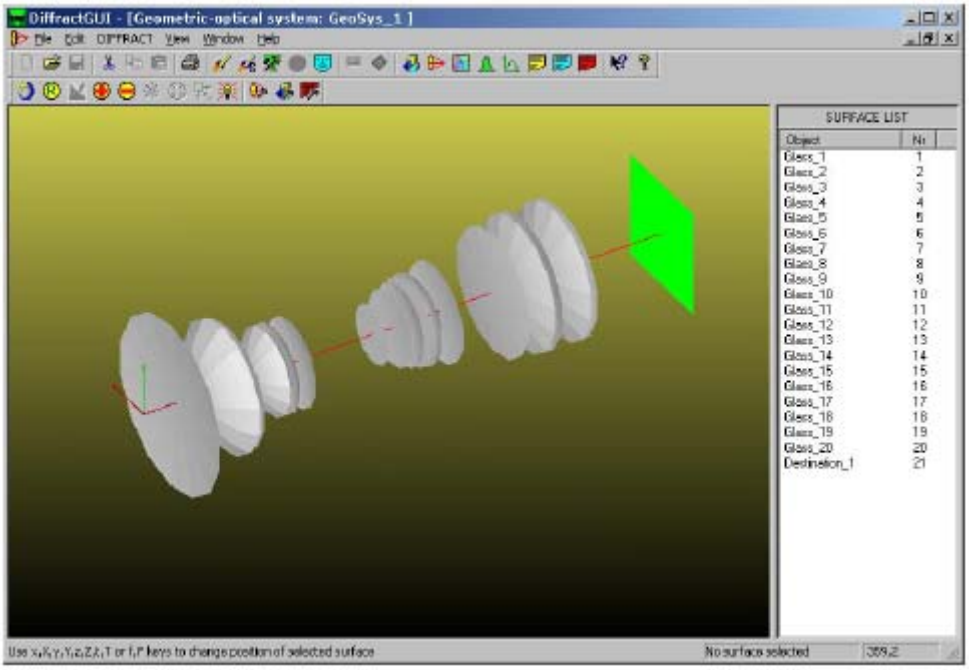
2 つの"…"ボタンで他の*.zmx ファイルを開くことが出来、またガラスカタログ（Cat list）への path を開くことが出来る。導入の間には、Cat リストは*.zmx ファイルに使われているガラスリストで満たされる。その 1 つを選択すると、Glas list にはそのカタログに含まれるすべてのガラスで満たされる。一つのガラスを選択すると、primary wavelength に対する屈折率が計算され、テキストボックスの中に示される。これは屈折率が対応しているかを直接実証するのに有効である。MIL 番号で表示した屈折率は DiffractGUI では扱っていない。

*.zmx ファイルは primary wavelength のみを考えている。この波長は、ファイルで使われている単位にしたがって、常に mm に変換される。

*.zmx におけるすべての光学面は表面データリストに表示される。これらは Zemax レンズエディターで用いられるリストと似たような外観を持っているが、編集する能力を持っていない。Surface-Data_List から 1 つの光学面を選ぶと、Info ラインはこの面の Extra Parameter を表示する（urface_Data_List にはそのためのスペースはない）。

*.zmx ファイルで表示されるすべての座標と方向は GEOP システムに変換され、Log アウトプットにリストされる。

Import ボタンをクリックすると、*.zmx 光学面を GEOP システムに変換をはじめる。Log アウトプットはこの過程における警告やエラーを追跡する。前の図に示した例について、その結果を下の図に示す。



一般に、第 1 面は省略される。これは Zemax™ では DIFFRACTTM に対応する object をもっていないからである。また、半径（D/2）が 0 にセットされた光学面も省略される。最後の面は通常 GEOP の最終面に変換される。

標準、傾いた光学面、奇数の非球面、および偶数の非球面は制限なしに受け入れられる。トロイダル面（toroidal）は Rx パラメーターと非球面係数が共にゼロの場合、円筒面として正しく受け入れられる（もしこれらのパラメーターがゼロで無い場合、トロイダル面は導入のとき無視されるが、警告は出る）。Alternate Even および Alternate Odd 表面は、警告が出た後、Even Asphere および Odd　Asphere タイプとみなされる。他のすべてのタイプの光学面は Standard 光学面として導入され、適当な警告が出される。DIFFRACT™ で使うために、それぞれの導入された光学面の開口半径は、対応する Zemax™ の半径と等しいとされる。Vignetting, 種々の開口マスク、および他のすべての Zemax™ の独特のパラメーターは無視される。

7.11 マウスホイールによる光学面パラメーターの変更（Modify Surface Parameters with the Mouse-Wheel）

Edit¥DIFFRACT_Parameters dialog box は単一光学面の変更には有用であるが、多数の面の変更には実用的ではない、すなわち、面のブロックを同時に変更するには適しない。このような場合、特別な d ialog box、the Modifier dialog が GeoSysView に備えられている。これは次の図で示すように、popup menu の Modifier...で起動できる。この modifier dialog は光学面のリストを備え、それらはマウスホイールの回転、または DIFFRACT_Parameter dialog box と同様に、リストに値を入れることにより、直ちに変更できる。マウスホイールのクリックのたびに、現在の値から新しい値に次の法則に従って変更できる。

$$NewValue = OldValue \cdot ((Factor - 1) \cdot WheelFactor + 1) + Offset \cdot WheelFactor$$

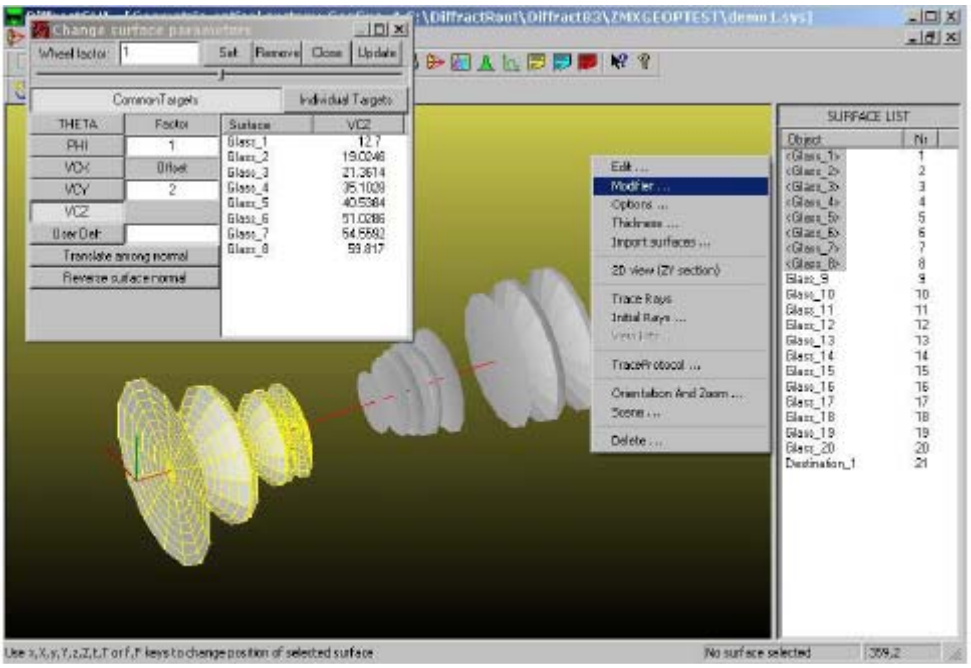
または

$$NewValue = (OldValue - Offset \cdot WheelFactor) / ((Factor - 1) \cdot WheelFactor + 1)$$

これらの選択は、ホイールの回転方向によって決まる。Modifier dialog は次の 3 種のオペレーションモードを持つ。

- Common Targets　Common Targets ボタンによって起動する。
- Individual Targets　Individual Targets ボタンで起動する。
- Deactivated　Common Targets および　Individual Targets の両者を押さないこと。

Dialog box が deactivated の場合、マウスホイールは GeoSysView のディスプレイを拡大・縮小するために用いる。



Common targets モードでは、modifier は異なった光学面のリストと、変更するパラメーターの名前（たとえば VCX, VCY, VCZ など）を表示する。このモードでは、すべての表面についてただ一つのパラメーター（すなわち共通のもの）のみを選択できる。この共通パラメーターの名前は、前に定義された **THETA, PHI, VCX, VCY,** あるいは **VCZ** チェックボタンで選択できる。もし、**USER Def:** ボタンをチェックすると、他のパラメーター名（NREFRACT のような）を直接エディットフィールドの中に入れることが出来る。Factor と Offset は新しい値を対応する編集画面に入れればよい。入力を有効とするためには Update ボタン（または Enter ボタン）をクリックすればよい。

Translate among normal オプションでは、ホイールクリックごとに目的とする光学面をそれぞれの面法線ベクトルに沿って移動させる。このモードでは、単に Offset のみを考えていて、Factor は無意味である。Reverse surface normal ボタンをクリックすると光学面の法線ベクトルを 180°変更するが、頂点の位置の座標は変更しない。

特定の光学面のパラメーターはリストで連続的に変更できる。リストを左ダブルクリックすると、ユーザーは異なった値を入力できる。しかし、その後で新しい値を適用するためには、Set ボタンをクリックしなければならない。そうしないと次の update のときに古い値が入力されてしまう。**Remove** ボタンを押すと、光学面は target list から削除される。**Set, Remove** の両ボタンは、もしあれば特定の選択した光学面に適用すべきである。そうでなければ、リスト全体が消えてしまう。

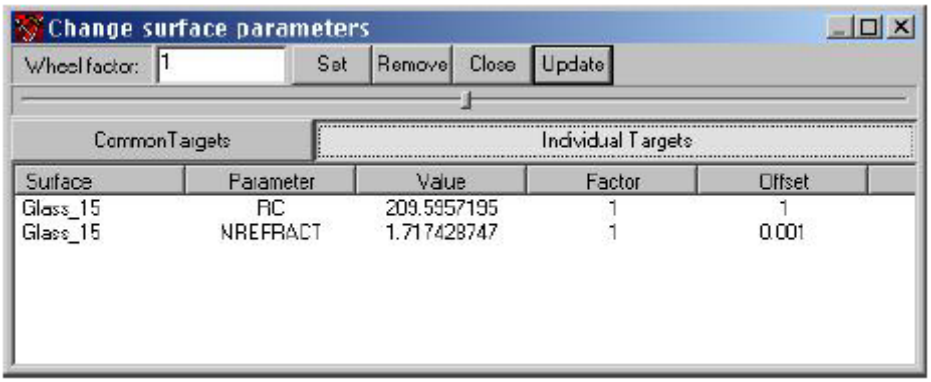
どのような GEOP 面でも Modifier dialog の common_target_surface_list で加える（あるいは取り除く）ことが出来る。これは GeoSysView（あるいは GeoSysView の右側にある SURFACE LIST）で **Curl** キーを押しながら直接希望する光学面をクリックすればよい。Common_target_list に含まれている光学面は GeoSysView の中に表示され、黄色のメッシュで覆われ、その名前は SURFACE LIST の中に<and>で囲まれて表されている。

Common_target_surface_list を設定する他の簡単な方法は、GeoSysView の光学面に矩形をマウス右ボタンでドラッグ（ボタンを押し下げながらマウスカーソルを動かす）すればよい。（もし何も面にタッチしないと、common_target_list は空である。このようにして作られた GeoSysView の背景を空にするには、小さな矩形をドラッグすればよく、common_target_iist をクリアーする便利な方法である）。

Individual Targets オペレイティングモードでは、次の図に示すように、リストにあるそれぞれの光学面に対し異なったパラメーターを選ぶことが出来るような、さらに一般性が許される。**Common targets** モードに対しては、1 つの光学面は何度もリストに現れ、このよ

うにして同じ光学面のパラメーターを逐語的に変えることが出来る。最後に、ターゲットパラメーターのそれぞれに対し、個々の **Factor** と **Offset** を設定することが出来る。

ターゲットリストは最初は空である。新しいターゲットを加えるには、希望する光学面に対して Edit DIFFRACT パラメーターdialog box を開き（その表面に対して左ダブルクリックをするか、SURFACE LIST でその名前を左ダブルクリックする)、**Ctrl** キーを押し下げて Edit DIFFRACT parameter dialog box で希望するパラメーターを左クリックすればよい。下の図の例において、曲率半径と Glass_15 の屈折率がこの方法で追加されたのである。



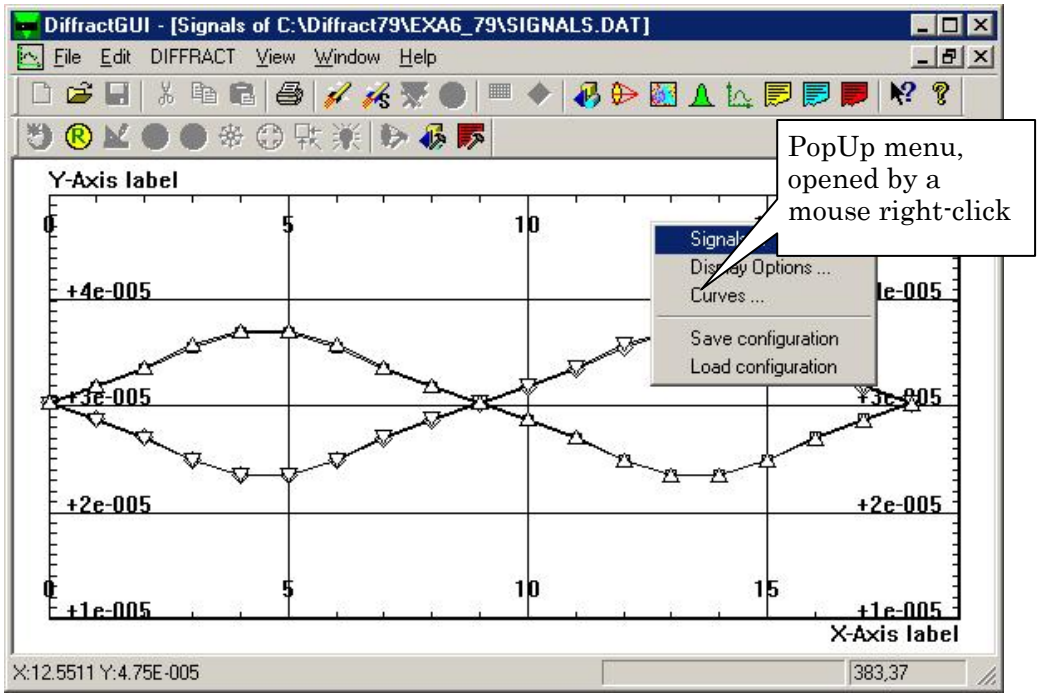
Factor と Offset はすべてのパラメーターについて 1 に初期設定される。これらを変えるには、リストの入力をクリックし、現れる edit field に新しい値を入力すればよい。同じように、パラメーターの値またはパラメーターの名前も変えることが出来る。**Common Targets** モードと同様に、**Individual Targets** リストも連続的に変化できる。したがって値を手動で変化するときは、希望する変化を入力してから **Set** ボタンをクリックすればよい。そうでなければ、次のアップデートのとき上書きされてしまう。さらに、不適なパラメーター名が指定されているときは、それらは削除されてしまう。

Common Targets モードと同様に、**Remove** ボタンはすべての、あるいは選択したパラメーター（もしあれば）を target list から削除するのに使われる。

8. SignalView

この View では DIFFRACT の SIGNAL.DAT ファイルがグラフィック表示できる。それぞれの信号が抽出され、曲線として表示される。ディスプレイや曲線の種々の様相は容易にしかも大幅に変更することができる。**SignalView** を開くと、*DiffractGUI* は現在使っている COMMAND,DAT と同じディレクトリに存在する SIGNAL.DAT を開こうとする。もし

そのようなファイルがない時は、そのことがユーザーに知らされ、何かデフォルトデータが表示される。

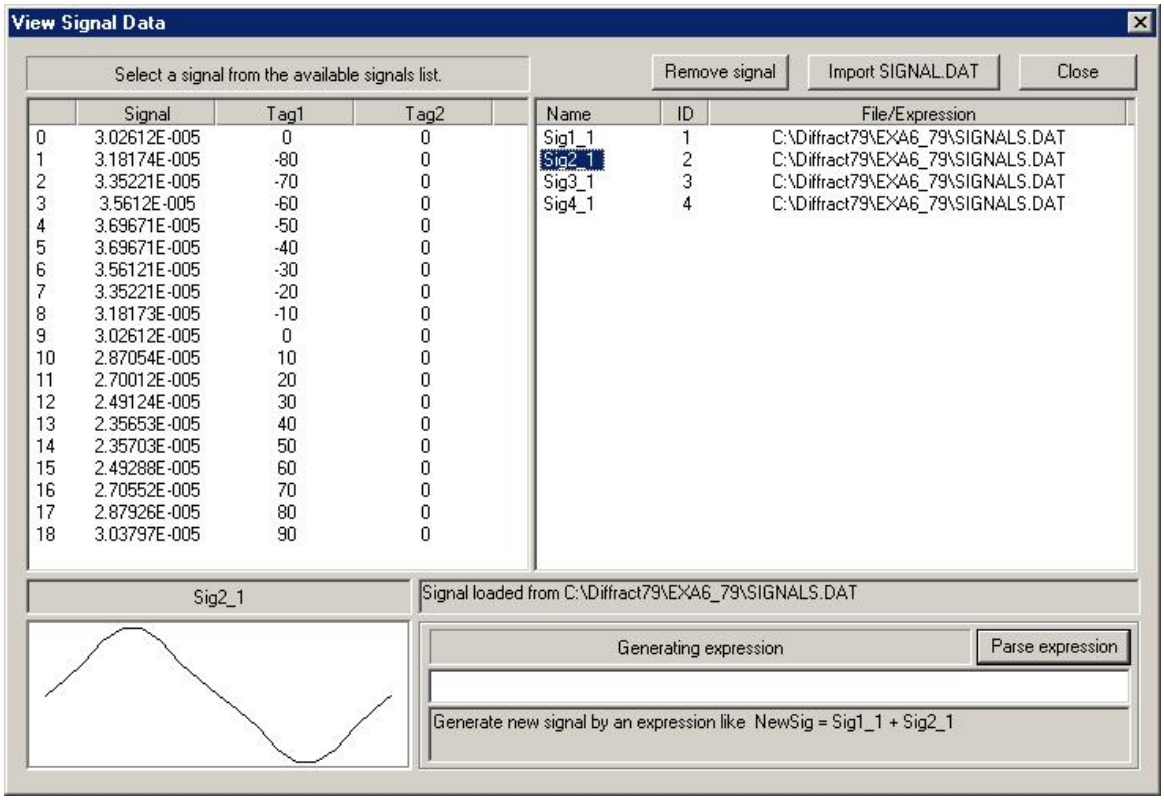


SignalView が起動しているときは、いつでも menu から **File-Open** を用いて、新しい SIGNAL.DAT ファイルを開くことができる。（開かれたファイル名は SIGNAL.DAT とは違うかもしれない。）SIGNAL.DAT ファイルの内容は変更することはできないが、新しいデータセットを作るとか、他の SIGNAL.DAT から加えることができる。曲線は何本でも表示できる。それぞれのカーブについてカーブの x および y 値に相当する 2 つのデータセットがある。ユーザーは種々のカーブについて、そのデータセットを特定すればよい。

マウスの右クリックで pop-up menu が現れ、信号を数値表で表示したり、ディスプレイパラメーターを変更することができる。Pop-up menu から **Curves** を選択すると、ユーザーは新しいカーブをディスプレイに追加することができ、あるいは現存しているカーブの様相を変更することができる。また、**Curves** のデータをファイルに保存することができる。

8.1 Signal Data の取り扱い（Handling Signal Data）

この対話ウインドウは pop-up menu の **Signals...**をクリックすると起動する。抽出された信号が右側にリストされ、その中から 1 つを詳しく知るために選択することができる。選択された信号の値は、関連するする Tag と共に左側に表示される。



簡単な下見用グラフが左下に表示される。(右側のリストボックスに)表示されたそれぞれのデータセットには、Sig1_1 のように自動的に名前が割り当てられる。(その上でマウスをダブルクリックすると、この名前を変えることができるが、リスト内で重複しないように注意が必要である。) Sig1_1 は、“最初に SESSION.DAT からインポートされた ID 1 の信号”を意味する。 **Import SIGNAL.DAT** option により、ユーザーは他の SIGNALS.DAT を data set pool へ読み込むことができる。

SIGNAL.DAT ファイルが多数の信号（100 まで）を含んでいるが、その中の幾つかしか必要がない場合がある。使われない信号（データセット）は、それを選択し（多数を選択することも可能)、**Remove signal** をクリックすればリストから削除することができる。信号はリストからだけ削除でき、元のファイル SIGNALS.DAT は変更できない。

リストに表示したそれぞれの信号（データセット）に対して、リストに表示されない 2 つの他のサテライトデータセットがある。これらは **Tag1Ofxxxx** と **Tag2Ofxxxx** といい、**xxxx** はリストに表示されているデータセットの名前である。たとえば、Sig1_1 データセットには Tag1OfSig1_1 および Tag2OfSig1_1 が付属する。

Dialog ウインドウの右下部は **Express Parser**（関係表現式）のための input box である。この parser によって、ここにタイプした代数的表現からスタートするどんな新しいデータセットでも作ることができる。新しい信号を作るために、parser の edit box に一般形の表現を入れる。

Left_Expression = Right_Expression

ここに **Left_Expression** は新しく作るデータセットの名前、**Right_Expression** は代数式である。この式の中に、リストに表示された信号の名前を変数として用いることができる。もしデータセットの 1 部分だけを使う場合は、(括弧の中に入れた)index の範囲を名前に付記する。新しいデータセットを作るためには、**Parse expression** のボタンをクリックするか、ENTER を押せばよい。

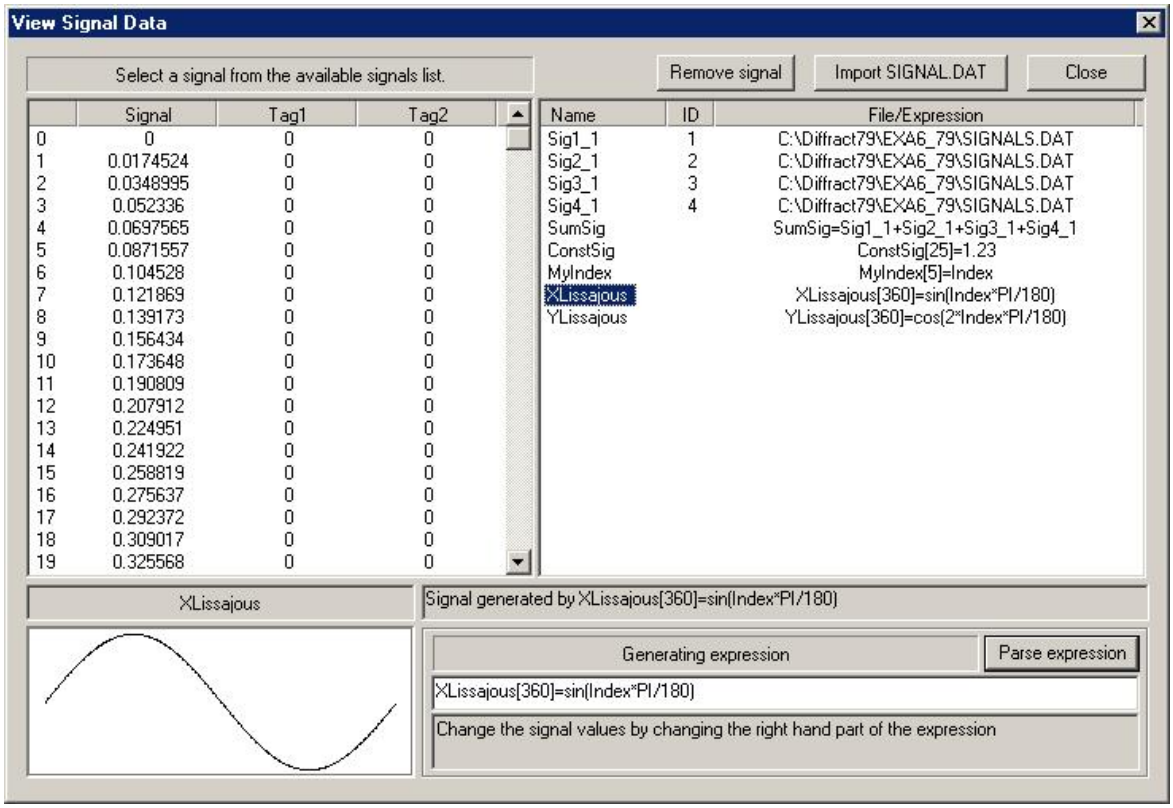
Right_Expression には次に示す数学関数を使うことが出来る：***sin, cos, tan, asin, acos, atan, dsin, dcos, dtan, sinh, cosh, tanh, exp, pow(x,y), ln,log, sqrt.*** （***dsin, dcos, dtan*** は ***sin, cos, tan*** と同じであるが、角をラヂアンではなく度で表す場合である。）更に、PI と EULER という語を数学的な定数 π と *e* の代わりに使うことが出来る。表現は point-by-point で行われ、最初の **index** でループプロセスを作ることができる。Parser は"*Index*"という言葉をこのループの index と置き換える。

註：ExpressionPaser は大文字・小文字を区別する。これは、たとえば、sin の代わりに Sin、Index の代わりに index、cos の代わりに Cos を使うと、構文エラーを生じる。

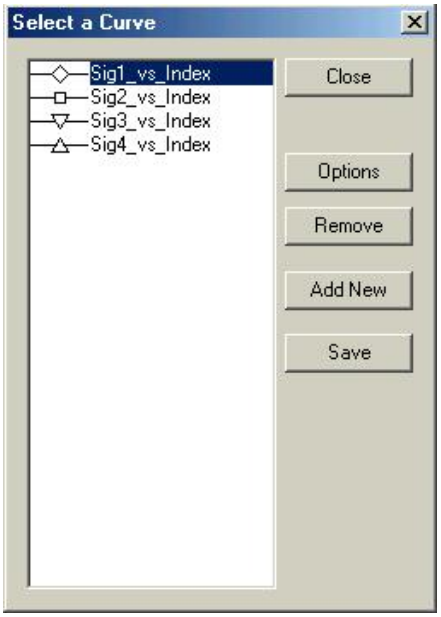
例：

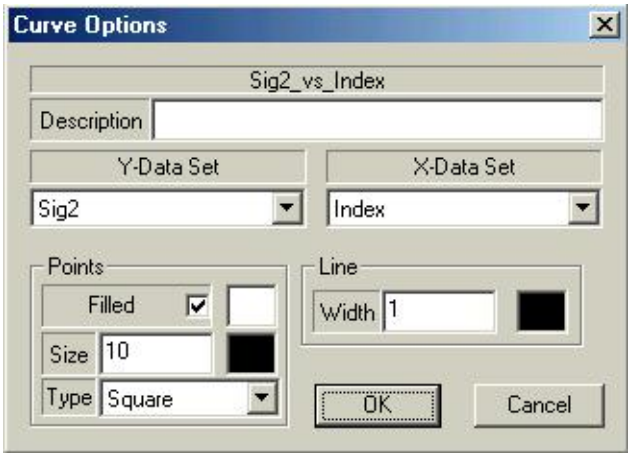
SumSig=Sig1_1+Sig2_1+Sig3_1+Sig4_1 は SumSig という新しい信号を作る。その値は Sig1_1, Sig2_1,Sig3_1 および Sig4_1 の和である。
***ConstSig*[25]=1.23** は ConstSig という新しい信号であって、25 個のデータ値がすべて 1.23 にセットされている。
MyIndex*[5]=*Index は MyIndex という新しい信号であって、5 個の値：0,1,2,3,4 からなる。
***XLissajous*[360]=*sin*(*Index*PI*/180)**は XLissajous という新しい信号を作り、0…360 度の領域で 360 個の sine 関数を含んでいる。
***YLisaajous*[360]=*sin*(2**Index*PI*/180)** は YLissajous という新しい信号を作り、0…720 度の領域で 360 個の cosine 関数を含んでいる。
SineOfSig1=sin(Sig1_1)は SineOfSig1 という新しい信号を作り、Sig1_1 と同じ数の信号を含む。SineOfSig1 のそれぞれの値は Sig1_1 の対応する値の sine である。

CopyOfSig2=Sig2_1 は CopyOfSig2 という新しい信号を作り、Sig2_1 と同じ値を含んでいる。



8.2 カーブの変更（Changing the Curves）



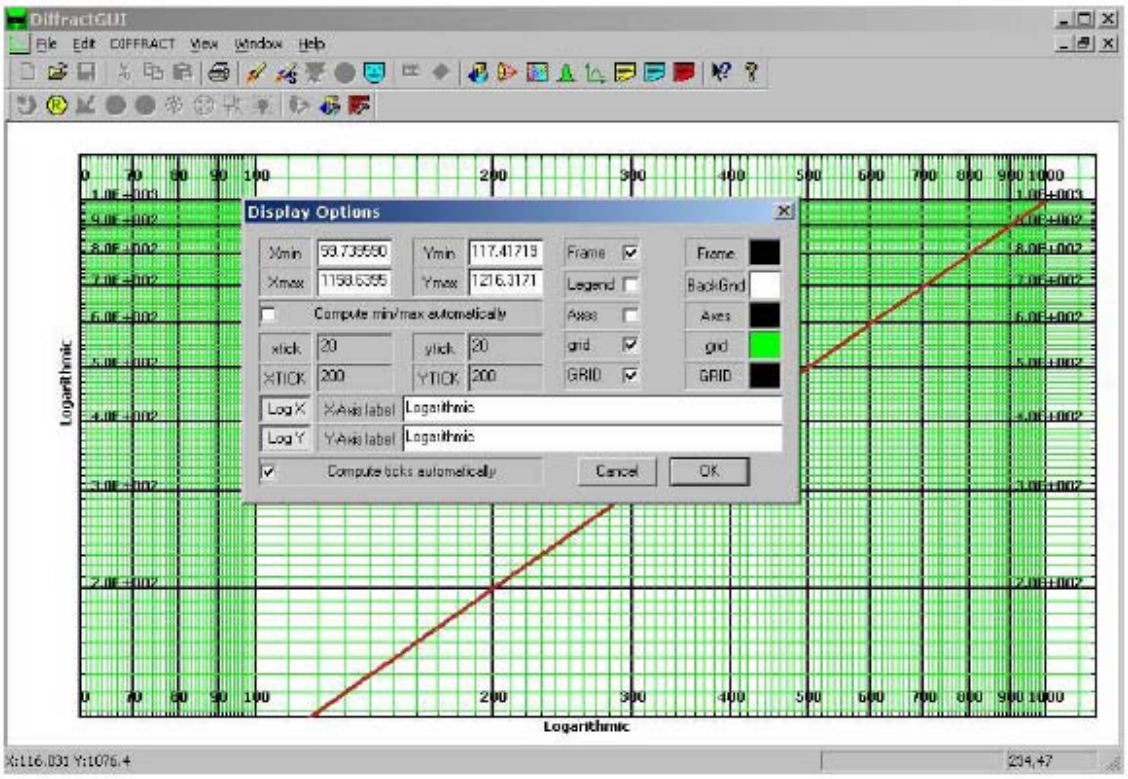


左上の対話ボックスは pop-up menu から **Curves** を選べば開かれる。すべての表示されたカーブがここにリストされている。これらのうち、１つのカーブを選ぶと、次の操作が可能となる。

- **Options**.　これは（右上に示した）**Curve Option** 対話ボックスを開き、カーブの幾つかの様相を変えたり、ｘ、ｙのデータセットを変更できる。
- **Remove:** データセットを変えることなく、カーブをディスプレイから消去する。
- **AddNew:** 新しいカーブを形成するため、新しいデフォルトカーブがリストに追加され、それに付属した **CurveOptions** 対話ボックスを開く。
- **Save :**　選択したカーブのデータセットを数値のペアーで表示したテキストファイルで保存する。

8.3　　ディスプレイの変更（Changing the Display）

pop-up menu の **Display options**...によってディスプレイの特性を変更することができる。



LogX および **LogY** のチェックボタンによって、display option dialog box は一つの軸の表示を他の軸に関係なく対数表示とすることが出来る。これは図に示してある。しかし、このようなオプションは $X_{\min} \le 0$,または $Y_{\min} \le 0$ のときは over-rule となる。そのときは、表

示は自動的に線形表示に戻る（Xmin または Ymin を大きくすると、0 は対数表示で再び機能する。）

8.4 Zooming and Panning

この機能は、DisplayOption 対話ボックスにおいて”Compute min/max automatically”のチェックボックスが動作しないときに限って用いられる。このボックスがチェックされているときは、下に示すユーザー入力はどれも動作しない。（すなわち、*DiffractGUI* は min/max 自動化によって、ユーザー入力を無効にする。）簡単な説明は次の通りである。

– ズーミングは左マウスボタンを押し下げたまま、マウスカーソルをスクリーン内で動かす。マウスカーソルを動かすと、スクリーンには矩形が表示される。マウスボタンを離すと、軸の min/max 値はこの矩形の値にセットされる。また、前の min/max 値は内部に保存され、グラフ上でマウスを動かさずに左ボタンをクリックすれば、復元される。最初のズーミングの後で、（zoom in zoom などを）続けることができるが、一連の操作の最初のズーミング値までしか復元できない。

– グラフをズームインまたはズームアウトするために、スクロールホイールを使っても良い。ズームの中心は常にスクリーンの中心であるため、貴方が見たい詳細までディスプレイをパンしても良い。

– パンを行うには、CTRL キーを押し下げ、左マウスボタン（ボタンを押して保持しながら）で drag すれば良い。パンの他の簡単な方法は、中央のマウスボタン（マウスのスクロールホイール：廻してはいけない、マウスボタンのように押し下げる。）で drag すれば良い。

– マウスの左ボタンをダブルクリックすれば、*DiffractGUI* は自動的に min/max 値を再計算する。これは、多くのズーミングやパンを行った後でスクリーンを復元するのに便利である。

8.5 Diplay Configuration の保存と読み込み（Save and Load Display Configuration）

SIGNAL.DAT ファイルから **SignalView** へ信号を読み込むと、DiffractGUI は自動的にその信号からグラフを生成する。そこで、カーブの様相を変更したり、あるいは SIGNAL.DAT ファイルから新しい信号を読み込んだり、代数的表現を使って新しい信号を作るなどの作業ができる。しかし、SignalView を閉じた後、これらの変更はすべて消失するので、それ

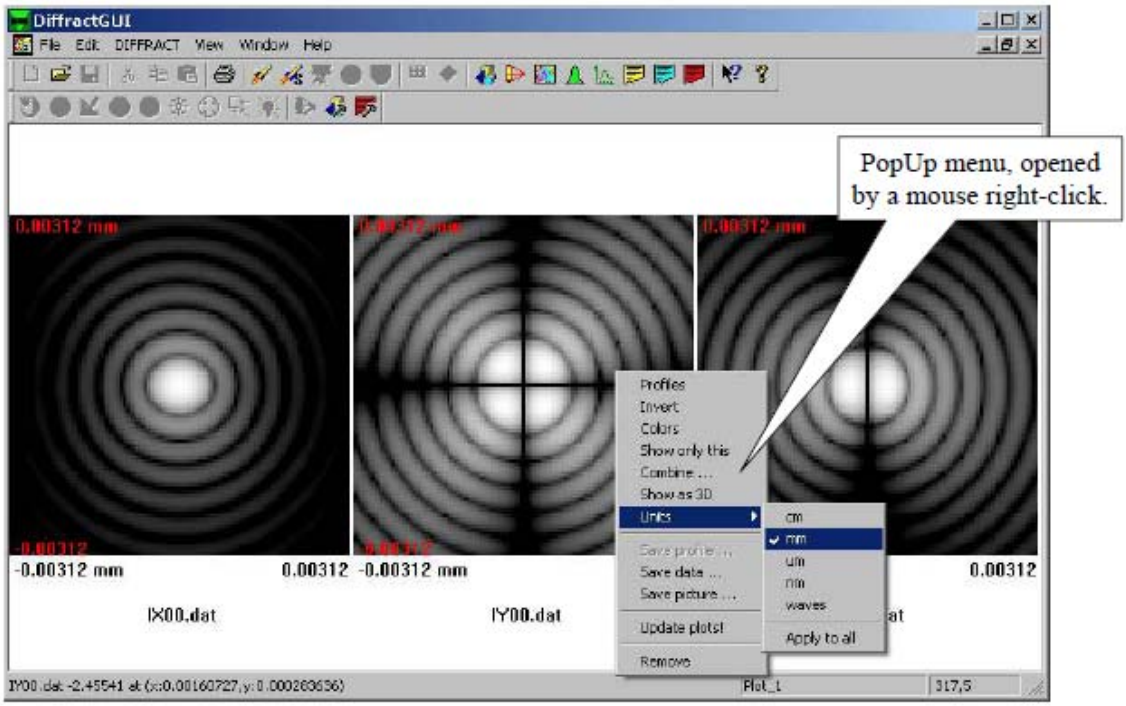
らを見たいときはいつでも手動で繰り返し実行しなければならない。このような面倒な操作を避けるために、現在の表示形式を configuration ファイルに保存し、後のセッションで読み込めばよい。この操作のためには pop-up menu の **Save configuration** と **Load configuration** を使えばよい。

DiffractGUI は **SignalView** に表示されたデータ（カーブ、データセット、expression など）に関するすべてのパラメーターを任意の名前のファイルに保存する。現実の信号（またはデータセット）は保存されないということを注意することが大切である。信号に関する参照情報(たとえば信号の ID、ソースファイル、生成 expression)のみが保存される。このファイルを再ロードすると、*DiffractGUI* はこの情報を認識し、元の object のディスプレイを再現する。

9. BitmapView

この view は、数字の列で表されるデータをカラーまたはグレイスケールの Bitmap 像に表現する。ファイルの中では、数字は少なくとも 1 つのブランクで分離されている。Bitmap の幅（ピクセル）は第 1 列のエレメントの数で与えられ、高さは列の数で決まる。すべての列は第 1 列と同じエレメントを持っている。それ以上の数は除外され、数が足りないところは 0 とみなされる。

DIFFRACT によって作られる出力ファイルはこの要求を満たしている。特に**.dat** という拡張子を持つファイル（たとえば、Ixnn.dat, Iynn.dat,など）がそれである。DiffractGUI の menu で **View-Bitmap** ボタン をクリックすると、対応するウインドウが開き、その後 **File-Open** を選び、希望するデータファイルを読み込む。一時に 1 つ以上のファイルを読み込むことができる。処理中の document(COMMAND,DAT)はこの処理の間、変化しない。



BitmapView が **Workbench** から起動され、**Workbench** から PLOT あるいは DISK object が選択されると、対応するデータファイルが使っている COMMAND.DAT のディレクトリから自動的に読み込まれる。（これはもちろん、DIFFRACT が前もって起動されていて、必要なデータファイルは既に作業ディレクトリに存在しているときに可能である。）この場

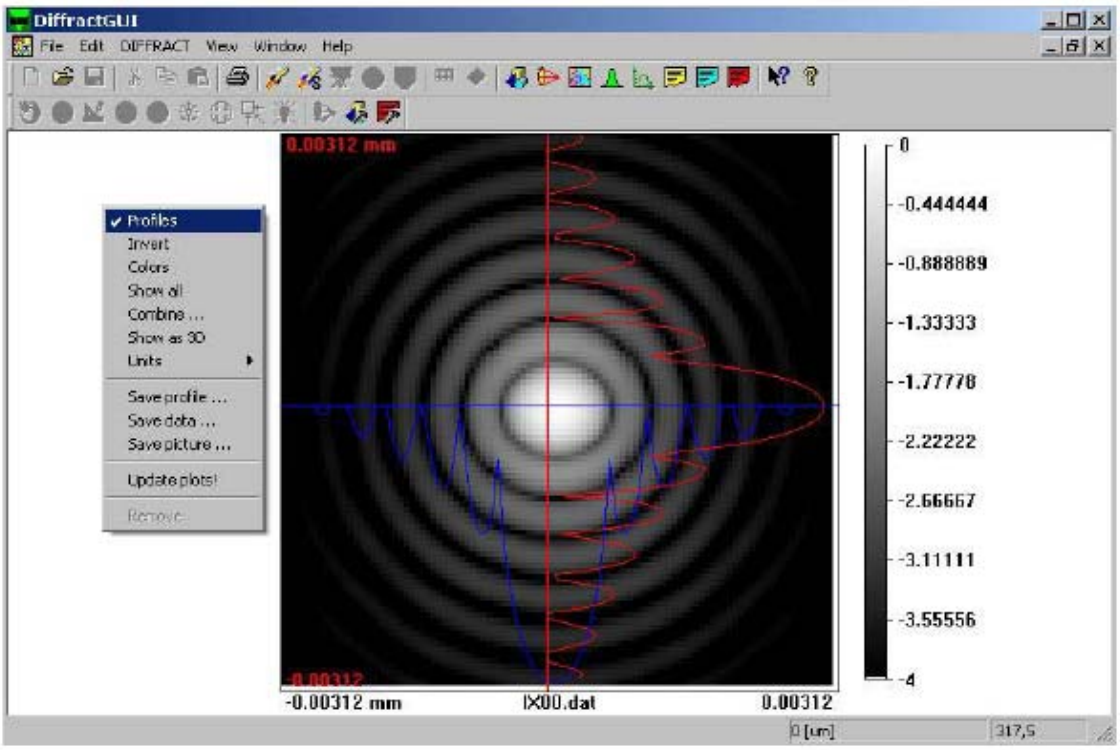
合、DiffractGUI は選択した PLOT-object のパラメーターから描画の物理的寸法を mm で決めることができる。

Pop-up menu の **Show only this** により、ユーザーは一度に 1 個の bitmap を描くことができる。これはいくつかの bitmap が使われるとき便利である、すべての bitmap を見るためには pop-up menu から **Show All** を選択すればよい。

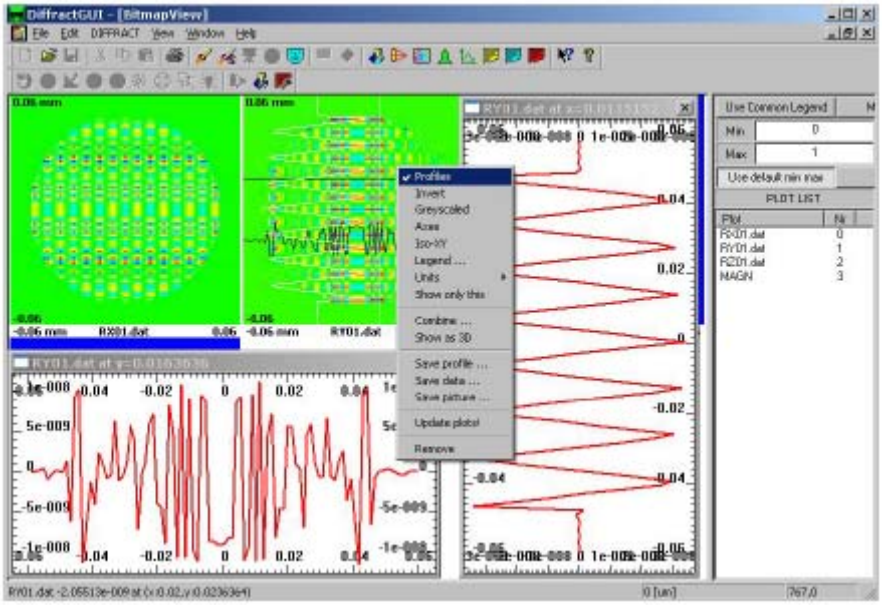
Combine dialog box は、Pop-up menu の **Combine…**で起動できるが、は、BitmapView に読み込まれたデータセットに基づいた新しい bitmap を作ることができる。

Edit field に表現を入れるときには、上の図に示すように、変数の名前として導入された曲面の正しい名前を入力する。表現は次のような形式で無ければいけない。Name = formula ここで Name は変化させようとする曲面 の名前である。(もし特定の曲面が存在しないときは、それが生成される。もしそれを保存したいときは直ちに保存しなければならない。) OK ボタンを押す(または ENTER を押す)とその表現は評価される。もしエラーが無いと、その表現は combo-list に付加され、後で再度使うことが出来る。前に使った表現のリストは DiffractGUI のセッションの間に記録される。さらに、それらは Clear Combo ボタンでテキストファイルに記憶され、テキストファイルから読み込むことも出来る (MaskEditor の表現リストと同じように)。エラーは combo-box の下の edit field(編集は出来ない)内に報告されるが、Clear Log ボタンで消去することも出来る。Combination bit map データは式で特定されたデータセットから構築される。もし combination bitmap が存在しないときは、新しい bitmap がメモリーの中に出来る。この操作によって何もディスクファイルが出来たり影響を受けたりしないから、ユーザーは結果を保存するためには pop-up menu の Save Data あるいは save picture を使わなくてはならない。**Save data** はデータを人間が読める ASCII ファイル (DIFFRACT と同じ) で保存する。**Save picture** はデータをウインドウズビットマップファイルとして記録する。

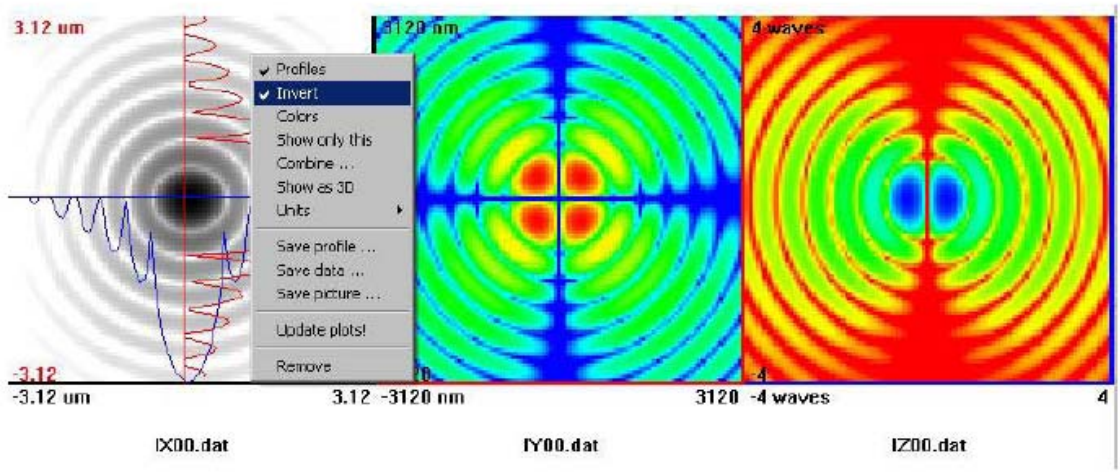
Pop-up menu の **Profiles** を起動すると、マウスの左クリックの度にディスプレイ中の強度分布のカーソルの位置における x-および y-断面が描画される。これらの断面図は pop-up menu の **Save Profile** を選ぶと、ディスクに保存することができる。



プロットの x - y 断面をよりよく表示するためには、次の図に示すように 2 つのサテライトグラフを付加する。これらのグラフは両者の断面曲線と、少なくとも一つの曲面が Profile オプションを可能にしていれば自動的に表示される。（どの曲面もこのオプションを持っていないときは、サテライトウインドウは自動的に閉じられる。）通常のウインドウのように、サテライトウインドウはスクリーンの周りに大きさを変えたり移動したりする。一時的にこれらが閉じられても、Profile オプションを持っている曲面をマウスクリックすれば再び現れる。1 つ以上の曲面が Profilles を持っているときは、最後にクリックした曲面の断面が表示される。



Bitmap 図の表示には 256 グレイ（カラー）レベルが用いられる；白（赤）はそのデータセットの最高値、黒（青）は最低値を表す。Pop-up menu の **Invert** を選ぶとこの関係が逆転し、“ネガティブ”像が現れる。

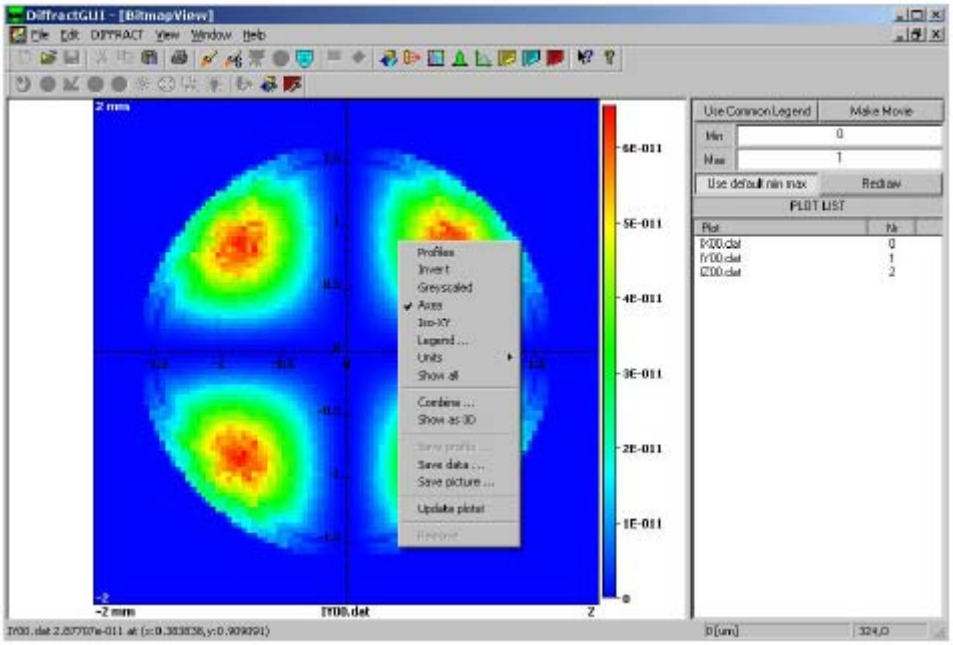


上図において、中央の bitmap は **Colors** option を持つが、右側の bitmap は **Colors** と **Invert** option の両方を持つ。この図はまた種々の長さの単位で描くことができる：それぞれの図には用いた単位を表示してある。これらすべての図をある特定の単位で表示するためには、pop-up-menu で **Unit-Apply to all** を選ぶと良い。

Show as 3D option は **ReliefView** ウインドウ（次章を見よ）で見ることができる。このウインドウではカーソルの下にある bitmap 像のデータファイルが読み込まれ、**ReliefView** ウインドウに 3D 描画が表示される。

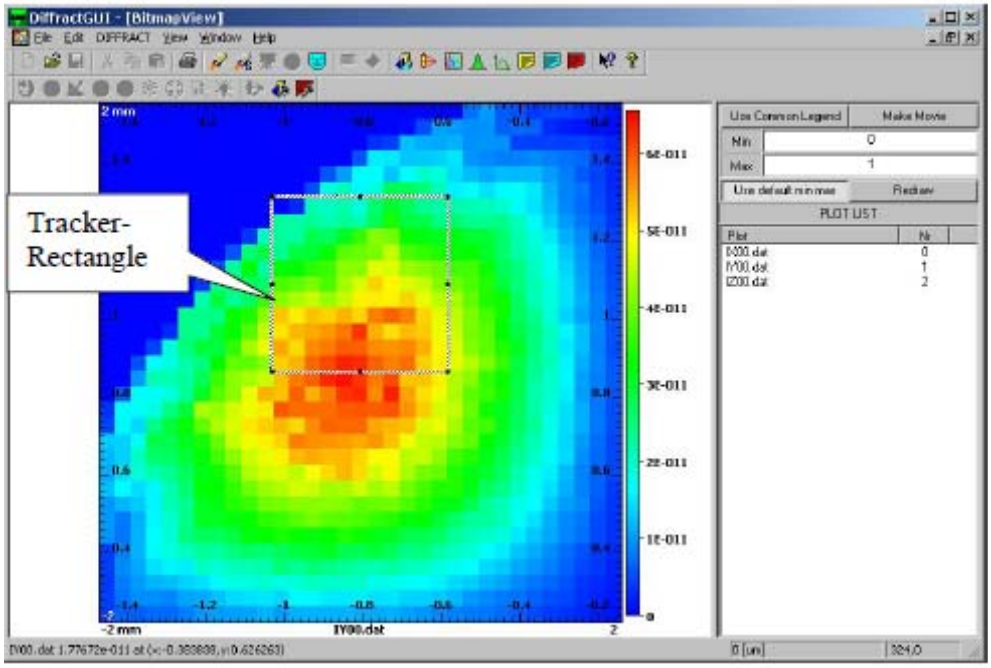
最後に、**Update plot!** Command は **BitmapView** にすべての描画ファイルを外部ファイルから再読み込みさせる。セッションの間にいくつかの DIFFRACT プログラムが実行されると、この特性はそれぞれのファイルの再読み込みを手動で実行することが必要となる。

BitmapView の右側にあるプロットリストは、View が単一曲面を表示しているとき（すなわち、pop-up メニューの **Show only this** が有効になっているとき）非常に有用である。このリストは現在 BitmapView にロードしたすべての曲面を含み、表示も可能である。リストから一つのアイテムを選び BitmapView をその曲面を表示するように設定する。それぞれの曲面はそれの種々の規定されたパラメーター、たとえば輪郭、カラー凡例、$z_{min} - z_{max}$ の拡張などで描画される。曲面リストは多数の選択が可能であって、1 つ以上の曲面が曲面リストから変更される。リストされたアイテムの右クリックは BitmapView の右クリックの popup による同じメニューを呼び出す。したがって、たとえば、一つ以上のグレイスケールモードの曲面をおくときには、リストから曲面の名前を選択し、選んだ名前の一つを右クリックすると、pop-up メニューから **Greyscale** を選ぶことが出来る。（リストから 1 つ以上の名前を選ぶときは、ウインドウズのやり方でキーボードの **Ctrl** または **Shift** キーを使えばよい。）



ZOOM と PAN　Bitmap View が single plot mode のときは、曲面を zoom あるいは pan する可能性がある。Zoom には、マウスのスクロールホイールを使うとよい。X 方向へ pan するときは **shuft** キーを押しながらマウスウイールを回すとよい。Y 方向に pan するときは、**curl** キーを押しながらマウスウイールを回すとよい。ディスプレイプロットにおいて

左ダブルクリックをすれば、初期寸法に戻すことが出来る。(**註**：曲面を最初のサイズより小さくすることは出来ない。ズーム特性は、下の図に示すように、拡大鏡のようなものである。)

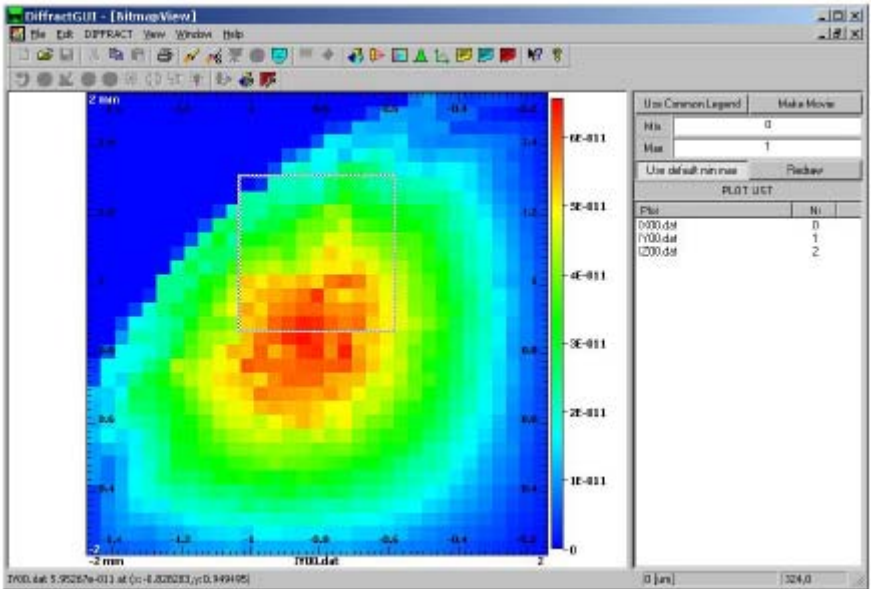


TRACKER-RECTANGLE　BitmapView の single mode における他の可能な特性は、矩形のサイズ変更であって、境界内のデータ点をピックアップして使うことが出来る。ピックアップしたデータ点は移動させ、サンプル曲面の他の部分にコピーすることが出来る。あるいはウィンドウズのクリップボードに（ASCII マトリックスとして）入れて、テキストエディターにコピーしたり、他の曲面、あるいは DiffractGUI の MaskEditor にコピーすることが出来る。

初期設定では Tracker Rectangle は起動できないし、見ることも出来ない。それを起動するには、キーボードの **Ctrl** キーを押しながら、マウスの左ボタンを押して（押しながら）マウスカーソルを動かす。この操作の間に Tracker Rectangle はゴムバンドのように表示される。左マウスボタンを離すと、Tracker Rectangle は上の図に示すように表示される。そのサイズは境界の隅にある小さい黒い矩形をドラッグすれば変えることが出来る。Tracker Rectangle の位置はカーソルを矩形の中に置き、左マウスボタンでドラッグすればよいが、プロット境界の外には出すことは出来ない。

この Tracker-Rectangle は元来空であって、何のデータも持っていない。この状態では、透明でサイズ可変である。Tracker をデータで満たすためには、マウスカーソルをその上に置

き、中央マウスボタン（通常マウスウイール）クリックすればよい。中央ボタンを再度押せば Tracker からデータは無くなり、再度空になる。マウス中央ボタンは tracker を“データ入り”と、“空”の状態を往復させることが出来る。

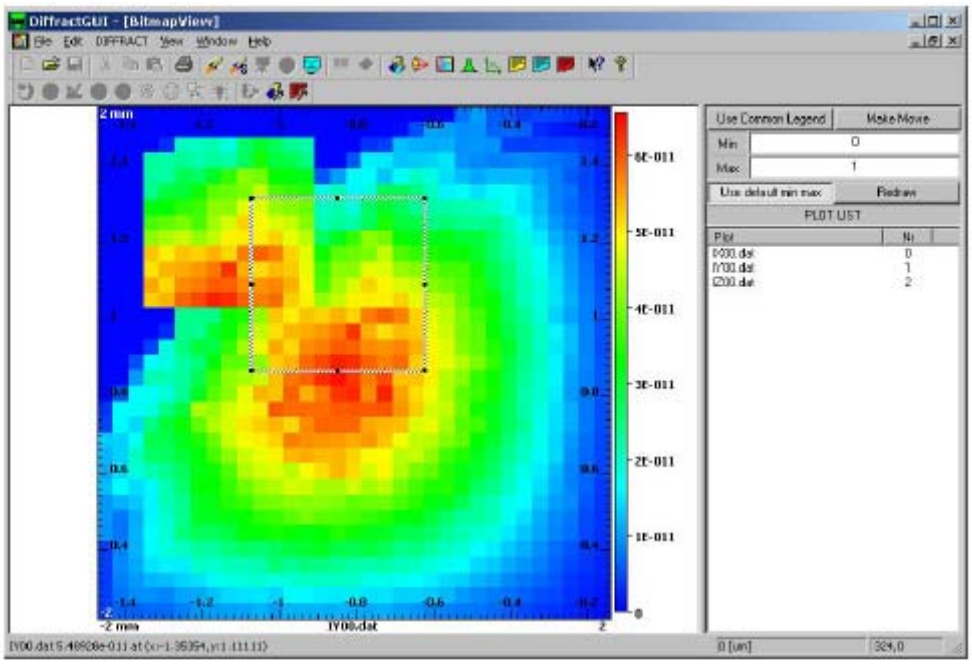


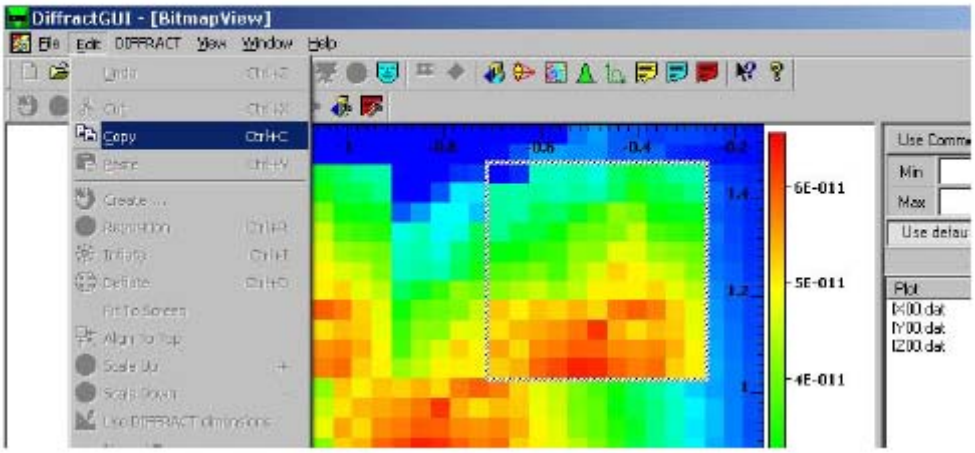
上の図は、下にあるフォトデータで満たされた Tracker の外観である。この状態を目で見て分かる唯一の差は、Tracker-Rectangle は境界にある小さな黒い矩形がないことから、もはやサイズを変えられないということである。またズーミングも情報入りの Rectangle では不可能である。しかし、曲線の pannning と Tracker の動きは可能なので、Tracker は曲線の境界の内部ならどこへでも動かすことが出来る。

一旦 Tracker がデータを持つと、その内容は単に **Ctrl** キーを押し下げながら Tracker の中央マウスボタンをクリックすれば、他の位置の曲線にコピーすることが出来る。（**注意**：この方法はプロットデータを取り消しできない方法で確認することなく上書きすることになる。）Tracker はコピーの後、データ入りの状態で残存することになる。このようにして、同じ内容を他の場所に同様にコピーすることが可能となる。（**註**：Tracker-Rectabgle はまた maskEditor でも使用でき、繰り返し特性またはマスク上に周期構造を作ることが出来る。）

Ctrl キーをを押しながら曲線の上でマウスの左クリックをすれば、Tracker-Rectangle を消し、スクリーンから取り除くことが出来る。（これは起動するときの動作と同じである）。BitmapView の曲線の一部を MaskEditor に移すときには、Tracker-Rectangle の内容を（Edit-Copy メニューコマンドを使って）クリップボードにコピーし、その後そのデータを MaskEditor に貼り付ければよい。次の図は Tracker-Rectangle のコピーと貼り付けの特性

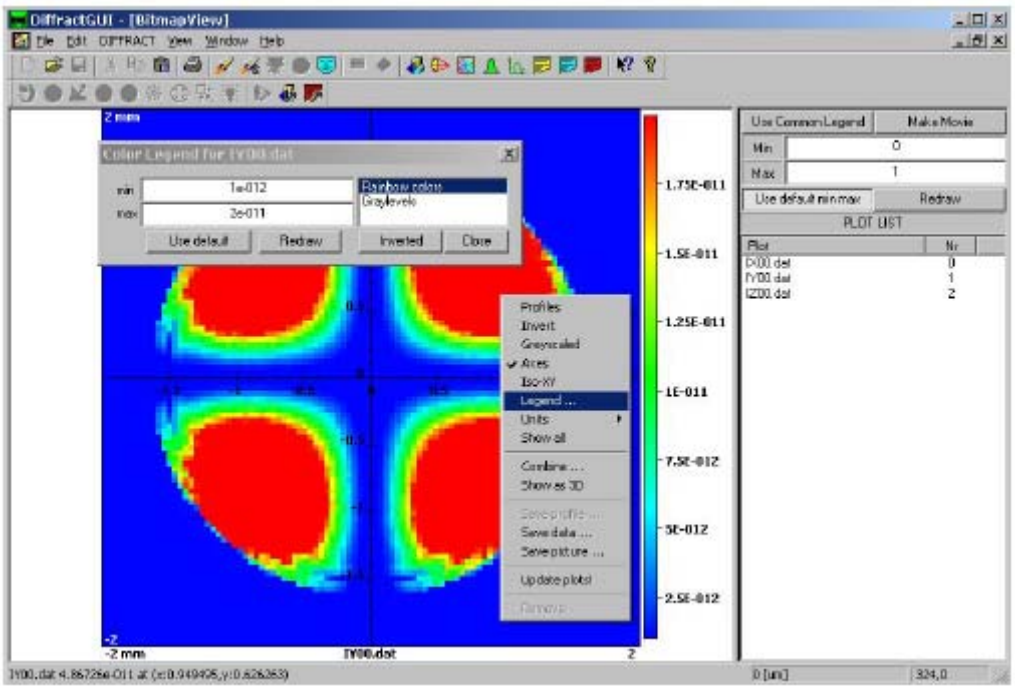
を示す。コピーコマンドは Tracker が情報を持っているときに可能であるが、貼り付けコマンドは Tracker が switch-off のとき（すなわちスクリーンに現れないとき）に可能である。





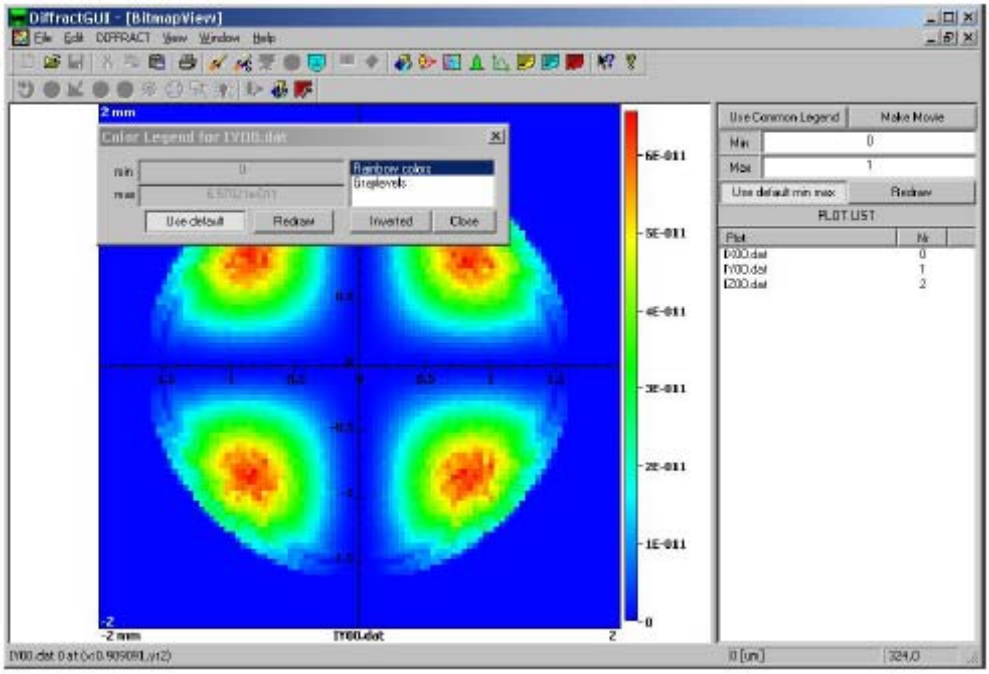
COLOR REGEND　Plot-data から plot-bitmap を作るとき使われる color regend はそれぞれの plot に対して color regend box を開く pop-up menu コマンド Legend....で実行できる。（下図に示す。）初期設定では、color regend の min-max 値は自動的に図のデータの min-max に設定される。このメカニズムを無効にするためには、Use default チェックボタンを選択せずに、dialog-box の対応する edit-fields に希望する min-max 値を入力すればよい。この dialog-box はモードレス、すなわち閉じることは出来なくて、1 つ以上の図を変更する場合は再び開かれる。簡単には dialog-boz を日炊いたままにしておき、BitmapView

の右側からリストから他の図の名前を選択すると、dialog-box は選択した図に自動的に更新される。新しい min-max 値を入力すると、効果を変えるために Redraw ボタンをクリックしなければならない。



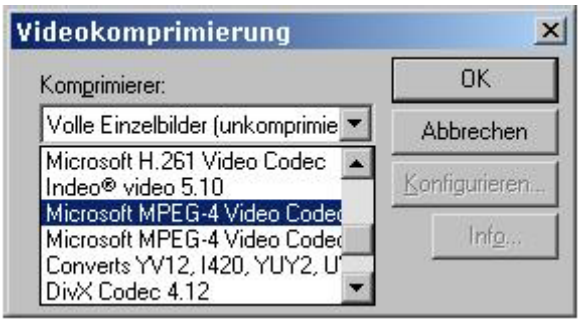
次の図は **Use Default** オプションをチェックしたときの color legend dialog box の外観を示した。Edit field はグレイになっていて、画像データの min-max 値がが入れられている。（この機能は曲面データの最低および最高値を入力するとき便利である。）　あらかじめユーザーが入力した min-max 値は失われることはない。それらは **Use default** オプションをチェックしなければ、edit-field に現れる。しかし、BitmapView を閉じるとすべてのそのような設定は失われる。

Dialog-box の右手のリストからカラーテーブルを”レインボウカラー”あるいは”グレイスケール “などと選ぶことが出来る。これは、pop-up メニューから” Colors”または “Grayscale” を選んだときと同じである。Color legend は **Invert** をチェックして反転することが出来るが、これは pop-up メニューの **Invert** と同じである。



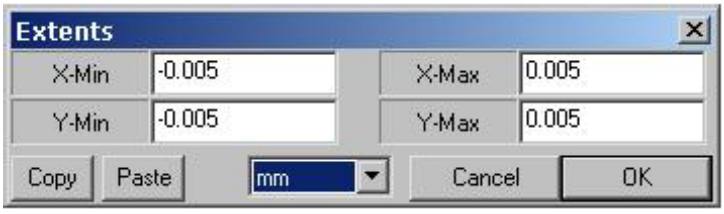
BitmapView の右側にある（plot list の上側）ボタンと edit field によって、個々の図の min-max 値は over-ruled になることがある。そのため同じ min-max 値がすべての図に適用される。このことを達成するため **Use Common Legend** ボタンをチェックする。Color legend dislog box なので、ユーザーが定義した global min max 値とすべての図に適用される min- max 値を入れ替えるのに、**Use default min max** ボタンが使える。

BitmapView に導入された prot から movie clip を作るために **Make Movie** を使うとき、すべての図のための common legend を使うことが有用である。スタートの前に、プロットリストから movie に取り入れることを希望する曲面の名前を選択したほうが良い。（もし、何も曲面を選ばなかった場合、すべての可能な曲面が使われる。）**MakeMovie** をクリックすると、DiffractGUI はファイル名を要求し（初期設定では＊.avi という拡張子を持つが、これは変更しても良い。）、その後使用する圧縮方法を要求する。（これはシステムメッセージで、言語は貴方の PC に使われているのと同じである。）



MakeMovie はシステムにインストールされている video 装置を使っているので、このオプションリストはコンピューターによって異なる。さらに、すべてのビデオコードは BitmapView に使われている bmp フォーマットでビデオファイルを作れるとは限らない。（疑いがあるときは、リストにある種々のフォーマットを試してみると良い。あるいは圧縮しない古典的な avi-file フォーマットを使うと良い。ただしこれはファイルのサイズがかなり大きくなるという不便がある。）

BitmapView あるいは ReliefView に表示される図の[Xmin, Xmax, Ymin, Ymax]値は Extent dialog box で任意に変更することが出来る。



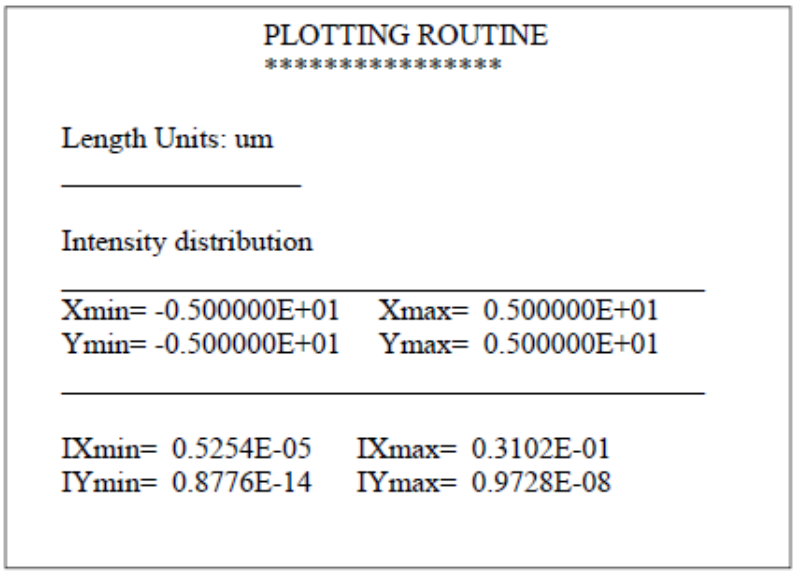
この dialog は pop-up メニューの Extent...コマンドで起動する。BitmapView ではこれを閉じると、BitmapView の右に表示された plot リストで選択されたすべての曲面に適用される。もし何も曲面が選択されないときは、pop-up メニューが最初に開かれた曲面の extent のみが変更される。

もし測定単位が変更されたときは、Extent dialog box の X-Min, X-Max, Y-Min, Y-Max は自動的に変更される。しかし図の表示に使った単位は変わらない。X-Min が変更されたときは、X-Max, Y-Min, Y-Maxha は自動的に追随する。（もし図に対して non-quadratic 表示が必要な場合、常に X-Min 値を最初に変更されたい。）

Copy と Paste ボタンは内容を Windows のクリップボード経由で伝達する。これは内容を session.dat ファイルでセットするとき特に有効である。その目的を達成するために、

SessionView を開き、下の図に示すように、長さの単位と Xmin, Xmax, Ymin, Ymax を含む文章の行を選ぶ。

SessionView の内部で、テキストブロックをクリップボードにコピー（Ctrl+C）し、その後 BitmapView に戻り、Extent dialog box を開き、Paste ボタンをクリックする。Extent dialog の貼り付け機能は、Length Unit を解剖し、Length Units:, Xmin＝, Xmax=, Ymin=, および Ymax= は entries であって、これらの entry の数値を取り込む。（これらは文法上または貼り付けられたテキストの長さについての厳格な要求ではない。しかし、Xmin＝, Xmax=, Ymin=, および Ymax=の前のブランクは文中に見かける IXmin=, IXmax=, IYmin=, IYmax との混乱を避けるキーワードである。）言うまでもないが、切り取り、貼り付けの機能は他の外部文書との export/import に使われたと考えられる。


PLOTTING ROUTINE
****

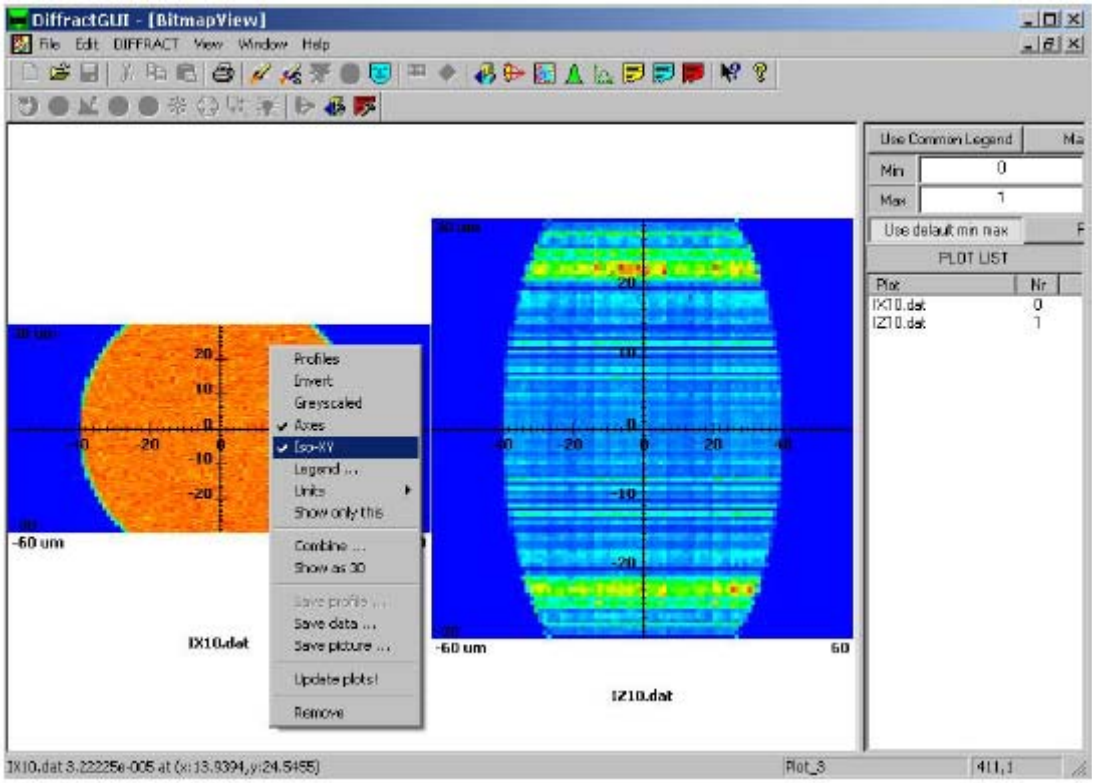
Length Units: um

Intensity distribution

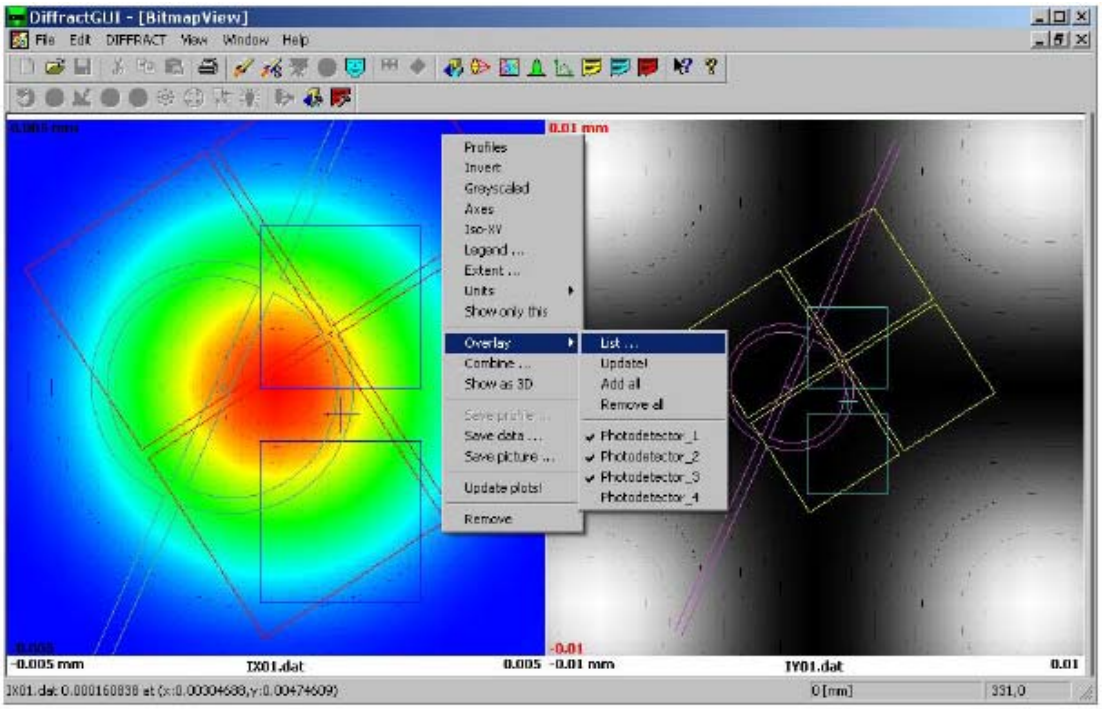
Xmin= -0.500000E+01 Xmax= 0.500000E+01
Ymin= -0.500000E+01 Ymax= 0.500000E+01

IXmin= 0.5254E-05 IXmax= 0.3102E-01
IYmin= 0.8776E-14 IYmax= 0.9728E-08

BitmapView はそれぞれの曲面に矩形の表示画面を与え、x および y 方向に個性的な広がりを持つ曲面に適している。しかし、x と y 方向の広がりが異なった場合、表示は歪んでしまう。この歪をなくするために、下図に示すように pop-up メニューの **Iso-XY** オプションを選択するとよい。図面の x 方向と y 方向は同じ広がりを持っているが、右側の曲面は **Iso-XY** オプションが使えない。**Iso-XY** オプションは曲線がメトリック単位の場合にのみ有効であり、**Axes** オプションについても同様である。



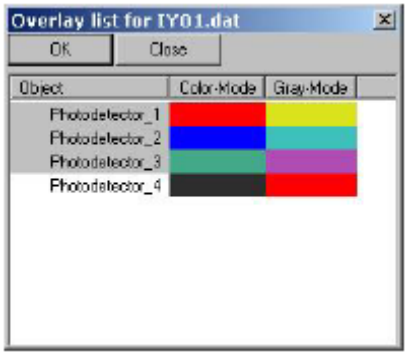
BitmapView と MaskEditor では、曲面は現在開いている command.dat で使っている photo-detector の outline-sketch を表示している。下の図はこれを BitmapView(MaskEditorView も同様)で表示したものである。



Pop-up メニューからはいる Overlay はこの特性に関連するいくつかのサブメニューをもっている。その中で最も重要なものはこの command.dat で定義したすべての photo-detector (DTCT)の実用リストである。それらのそれぞれはこのメニューリストを左クリックして overlay リストから取り入れたり、取り消したりすることが出来る。もし、photo-detector がこの曲面と共に表示されたら、その名前が pop-up メニューからチェックできる。実際には、メニューに表示される photo-detector の最大数は 20 に制限されている。もし command.dat ファイルに 20 以上の detector が含まれていても､最初の 20 個のみがこのメニューに表示される。

Add all と Remove all コマンドはすべての Photo-detector を取り付けたり、あるいは取り除いたりする。(20 以上の detector が定義されていて）それらがメニューに表示されていなくてもである。Update コマンドは曲面に関わったすべての photo-detector を再登場させる。これは、1 つあるいはそれ以上の photo-detector が Workbench で変更されていた場合に特に重要である。

List...は次のような dialog box を開く。



この dialog box は（メニューから）detector を図の表示リストから表示したり、取り除いたりすることが出来る。これは、ディスプレイされているリストボックスの名前の上で、マウスの左ボタンをクリックすればよい。取り付けられた detector はグレイの背景の上に表示される。

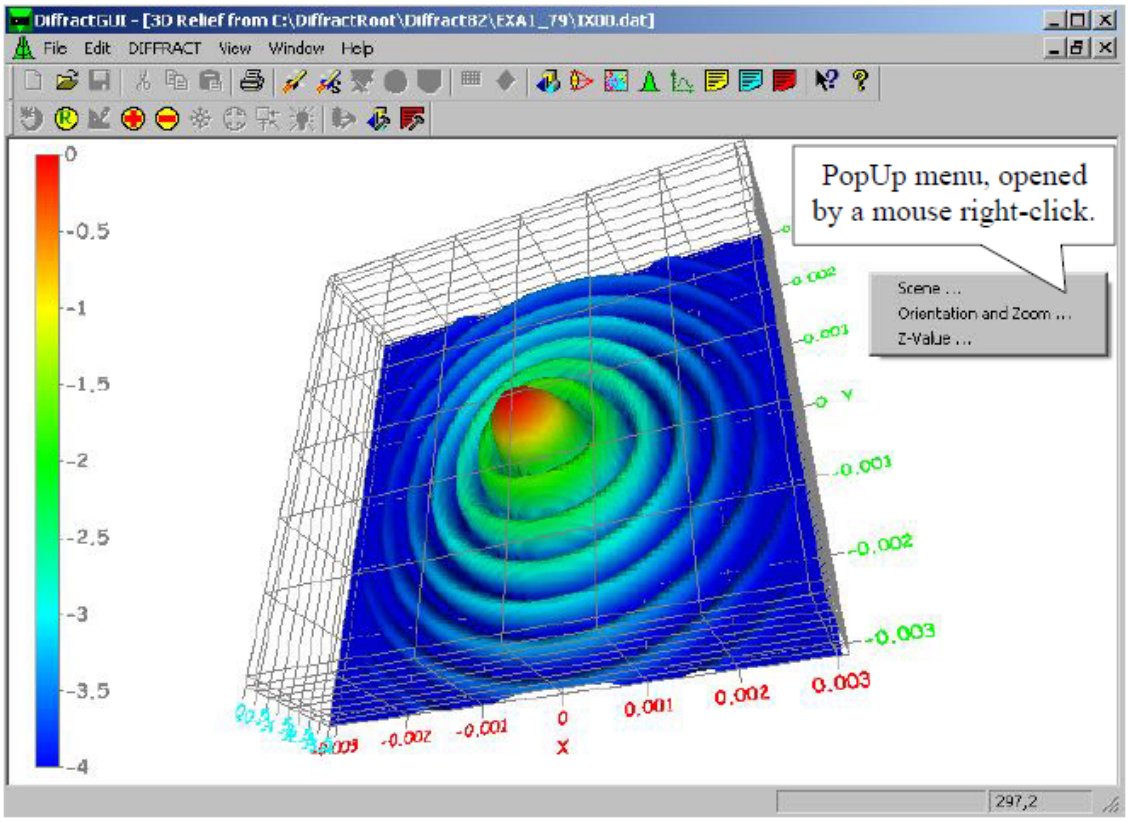
表示されたそれぞれの detector に対し、2 つの色が使われている。1 つは虹色カラーマップ（color-Mode コラム）を使って表示した曲面に対して detector outline を表示する。他の一つは曲面がグレイレベルで表示された（Gray-Mode コラム）ときに使われる。

元来、すべての Color-Mode カラーは黒で、またすべての Gray-Mode カラーは赤で示され

た。リストボックスに示された色の付いた矩形の 1 つを左クリックすると、標準の ChooseColors dialog box が開かれ、ユーザーにそれぞれの色を個々に変更することが出来る。それぞれの photo-detector に付けられた色は自動的に log-file と同じフォルダーに **pd_overlay_colors.txt** として保存される。そこから、DiffractGUI で command.dat が開かれると、その度に色は自動的に再入力される。もし、読み込みまたは保存処理が失敗しても警告やエラー表示は現れないが、代わりに初期カラーが使われる。

10. ReliefView

ReliefView は **BitmapView** に似ているが、それはデータを 3 次元グラフとして表示する。そのウインドウを開くには *DiffractGUI* menu から View-Relief をクリックする。このとき、**Workbench** で PLOT, SAVE, READ, DISK または MASK が選択されえていると、**ReliefView** は自動的に添付ファイルを読み込む。そうでなければ、あるいはファイルが開けないときには、ユーザーはファイルを開くように要請される。

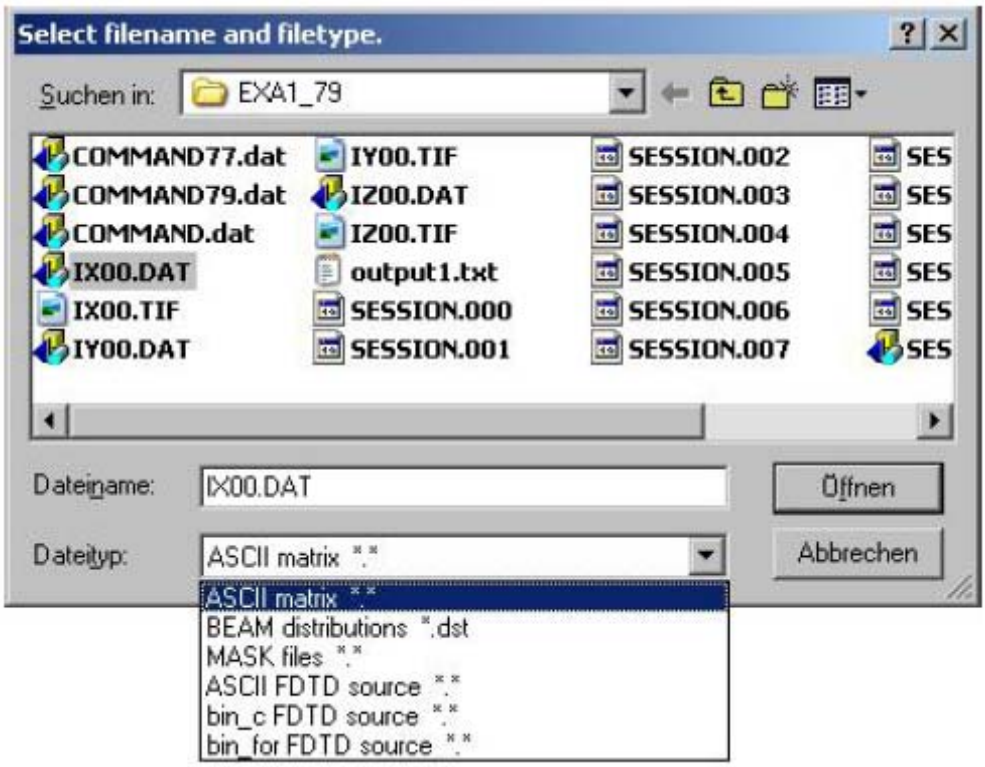


上の図は **ReliefView** の一般的な外観を示す。右マウスボタンをクリックすると pop-up menu が現れ、display option を変えるか、データ解析を行うか選択できるようになる。

10.1 ReliefView のためのファイルの読み込み（Loading a file for ReliefView）

DiffractGUI の **ReliefView** 内では、DIFFRACT の PLOT option で作られたデータを含む ASCII マトリックスファイルのみではなく、SAVE command で保存されたファイル（すなわち、ビーム分布ファイル*.dst）、外部 MASK データファイル、および DIFFRACT の export FDTD ファイルについても読み込んで表示することができる。

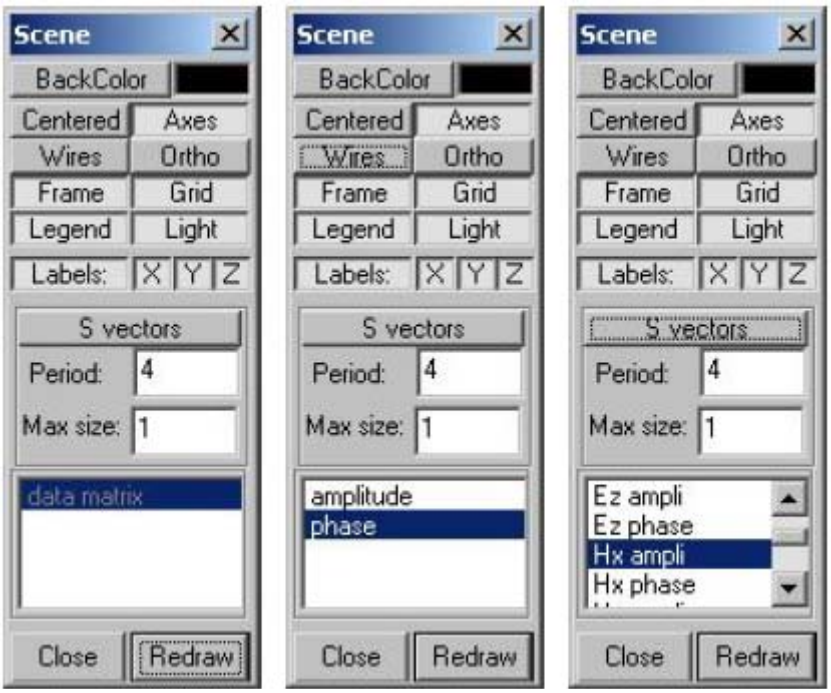
ReliefView が開くと、*DiffractGUI* menu command File-Open を用いて、表示したいファイルタイプとファイル名を選ぶ。次の図は **File Open** 対話ボックスからどのようにしてファイルタイプを選ぶかを示している。



一般に、まずファイルタイプ（ASCII matrix, BEAM, MASK, あるいは FDTD）を明らかにし、次にファイルリストから希望ファイルを選び、*Enter* を押す。表示は 1 つのファイルだけであることを注意されたい。新しいファイルが読み込まれると、古いデータは廃棄される。さらに、読み込まれたデータは **ReliefView** では訂正できない。したがって、ここからファイルを保存し戻すことはできない。

10.2 表示の option（Scene options）

Pop-up menu から **Scene** を起動すると Scene dialog box が開かれる。



上の 3 つの図は種々のデータタイプ（左から右に向かって PLOT, BEAM あるいは MASK, FDTD source）に対応する **Scene** dialog box である。PLOT で作られた ASCII matrix ファイルに比べて、BEAM, MASK, FDTD ファイルは表示すべきデータマトリックスが複数個存在する。BEAM(*.dst)と MASK ファイルは振幅と位相、FDTD ファイルは電場 E および磁場 H の各成分に対し振幅と位相がある。Scene dialog box の最下段に示されたリストから必要なデータマトリックスを選ぶことができる。

Scene dialog box では次のような option が可能である：

- **BackColor**：これをチェックすると、**ReliefView** の背景色が BackColor ボタンの右にある矩形の中の色に変更できる。この矩形をクリックすると、標準的な ChooseColor dialog box が開き、背景に違った色を選ぶことができる。もしチェックしなければ、デフォルトの OS 背景色が選ばれる。
- **Centered**：これをチェックすると、グラフは常にウインドウの中心に表示され、どのような移動の要求も無視される。
- **Axes**：スクリーンに直交 3 軸を表示する。（x 軸＝赤、 y 軸＝緑、z 軸＝青）
- **Wires**：描画を固体から wire- frame 表示に変更する。あるいは逆の操作を行う。しかし、メッシュの密度が高すぎると両者の表示に大きな差は出ない。
- **Ortho**：遠近法投影表示を正射投影表示に変える。この 2 つの表示の主な差異は、正射投影表示では深さの情報は考慮していない。

- **Frame**：平行六面体の枠が 3 次元表示 object を囲んでいる。レリーフパターンを囲んでいるグリッドと混同しないように注意されたい。
- **Grid**：この option はデータの周りをグリッド線で囲むか否かを切り替える。グリッド線は x, y, z 軸における ticks と同じレベルで描かれる。 z 軸 tick mark は color regend におけるものと同じである。
- **Legend**：カラー凡例バーをスクリーンの左側に表示するか否かを ReliefView に指示する。
- **Light**：表示されたグラフのコントラストを増加するため、光源（直接表示されてはいない）を活性化すればよい。もし光源を活性化すれば、表示された光学面のそれぞれのカラー強度は面から光源へ向うベクトルと面の法線との角に比例して変化する。
- **Labels**：3 軸のラベルは表示から削除することができる。もしこの option をチェックしなければ、x, y, z 軸のラベルはどれも表示されない。Option をチェックすれば、x, y, z 軸のラベルは **X, Y, Z** ボタンとは無関係に選択できる。
- **S vectors**：この機能は、FDTD データのみで有効である。**ReliefView** は電磁場の E および H 成分からデータマトリックスの各点における Poynting vector を計算する。もしこの option をチェックすれば、光学面は常に平面として表示され、方向を示す S ベクトルはグレイラインで表示される。**Max size** パラメーターは最大の S ベクトルの長さを決める。マトリックスの **period** 点のみに S ベクトルが表示される。

10.3　方向とズーム（Orientation　and Zoom）

Pop-up menu の Orientation and Zoom を起動すると、orientation and Zoom 対話ボックスが開く。その外観は図の通りである。

通常、スクリーン上における表示の方向はマウスの dragging によって変更することができる（マウス左ボタンを押しながら、マウスカーソルを動かす）。しかし、**X angle** と **Y angle** を決め（そして **Apply** をクリックするか、*Enter* を押す）れば、表示は回転する。一般的な見地から 6 個の角度を入力して行う設定は、**XY, -XY, XZ, -XZ, ZY,** および**-ZY** ボタンから実行できる。これらは、視点を 3 つの座標軸の一つにおいた場合を実現できる。さらに良い視点を選ぶためには、これらの視点を選んだとき、**Ortho** option をスイッチすることをお勧めする。

CTRL キーを押しながらマウスカーソルを drag すれば、スクリーン上で光学面を移動することができる。しかし、これは **Centered** option を不能にしたときだけ可能である。すぐ後に **Centered** をチェックすれば、前にどのような移動を行ったとしても、表示をスクリーンの中心に戻すことができる。

表示のサイズは **Scale** パラメーターで決めることができる。これは **UP** および **DOWN** をクリックして段階的に変更するか、edit ボックスに直接入力（そして **Apply** をクリックするか、*Enter* を押す）するか、あるいはもっと分かりやすい方法ではマウスホイールを回転すればよい。

大きなデータマトリックスでは画面の表示には長い時間（～100ms 以上）がかかるので、作動時間を少なくするために drag operation 中はマトリックスを粗く表示し、回転動作が終わったら full resolution で表示する。

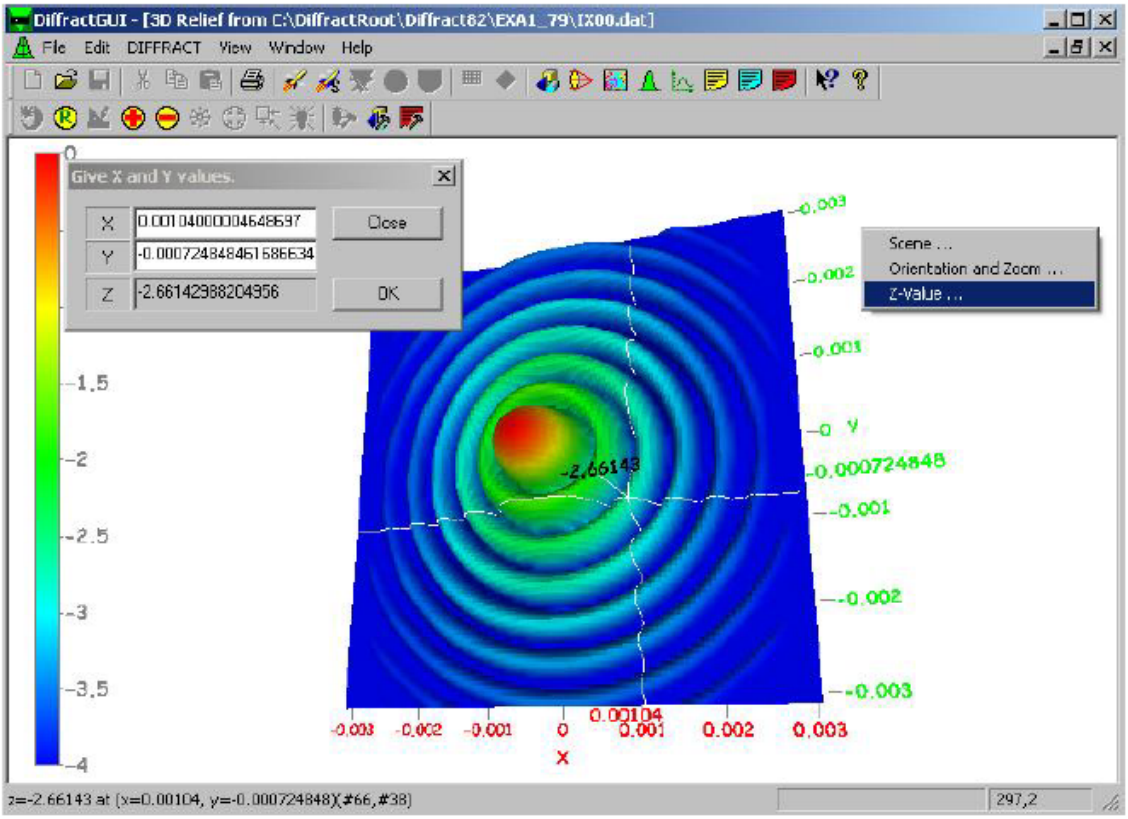
10.4　データ値の収拾（Picking data values）

表示されたデータをさらに精密に検査するために、光学面から x, y, z 値を抽出することである。もっとも簡単な方法はマウスカーソルを希望の点に置き、マウスの *Middle* ボタンをクリックする。もしマウスがホイールを持っているときは、ほとんどのマウスはホイールに Middle ボタンと同じ機能を持たせているため、ホイールを押す（回転させてはいけない）とよい。もし、中央ボタンもホイールも無いときは、SHIFT キーを押したまま左マウスボタンを押せばよい。

こうすれば、次の図に示すように、x, y, z 値はスクリーン上に表示される。さらに、直交する断面線が表面の凹凸に従って表示される。

光学面の x, y の位置での z の値を得るもう一つの可能性は pop-up menu の **Z-Value** を使う方法であって、x, y 座標を設定する対話ボックスが開かれる。*Enter* キーを押すたびに、あるいは OK ボタンをクリックする度に、与えられた(x, y)に対する z 値が対話ボックスの Z ボックスに表示される。

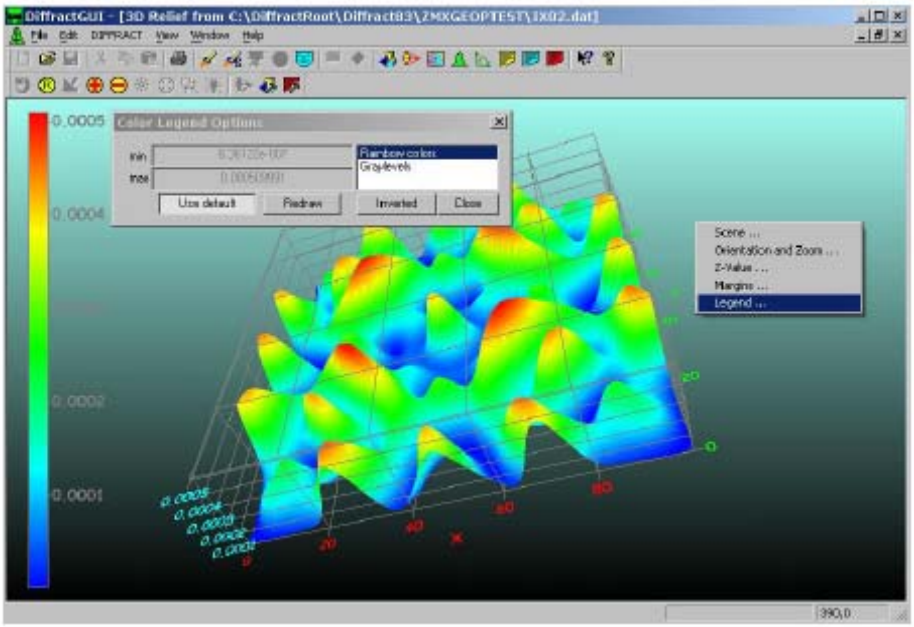
表示しているデータは離散的マトリックスから抽出されているため、すべての(x, y)位置に対する z 値を見ることはできないことに注意すべきである。*ReliefView* は要求された(x, y)座標を最も近いメッシュポイントに調整する。



x , y, z 値はフレームウインドウの status bar にプリントされている（上図を見よ）。補助的な情報として、その点の内部データマトリックスにおける行および列番号が status bar 上に現れる。

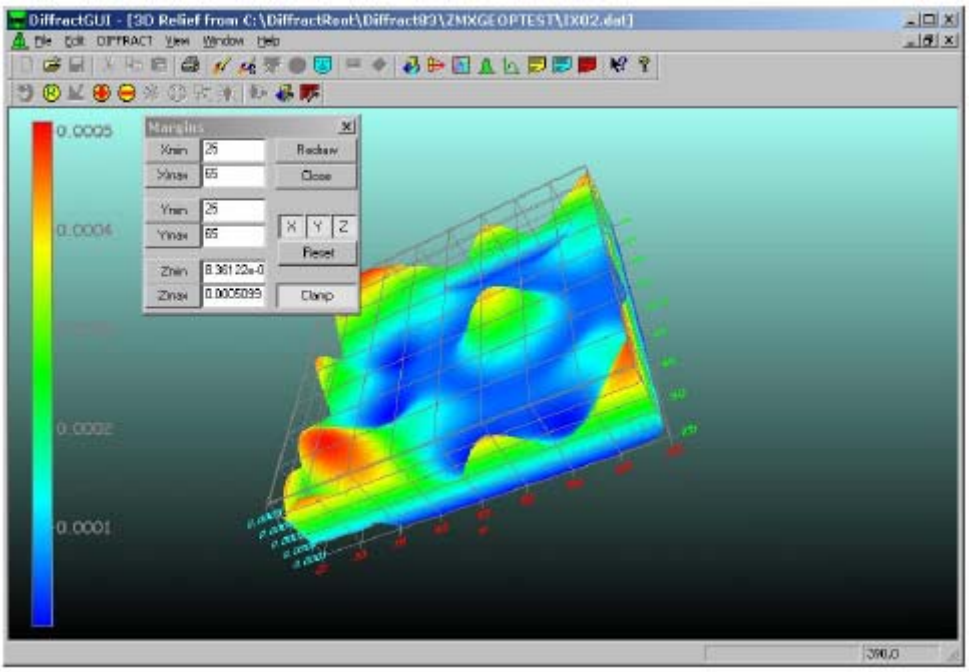
抽出された x, y, z の値および等高線は、表示部分の外側にある（x,　y）座標を選択して。スクリーンから消去することが出来る。

ReliefView の Color legend dialog box は、下に示すように、pop-up メニューの Legend… コマンドで起動できる。それは BitmapView の color legend dialog box と同様に働くので、さらに詳しいことはそのマニュアルを参照されたい。

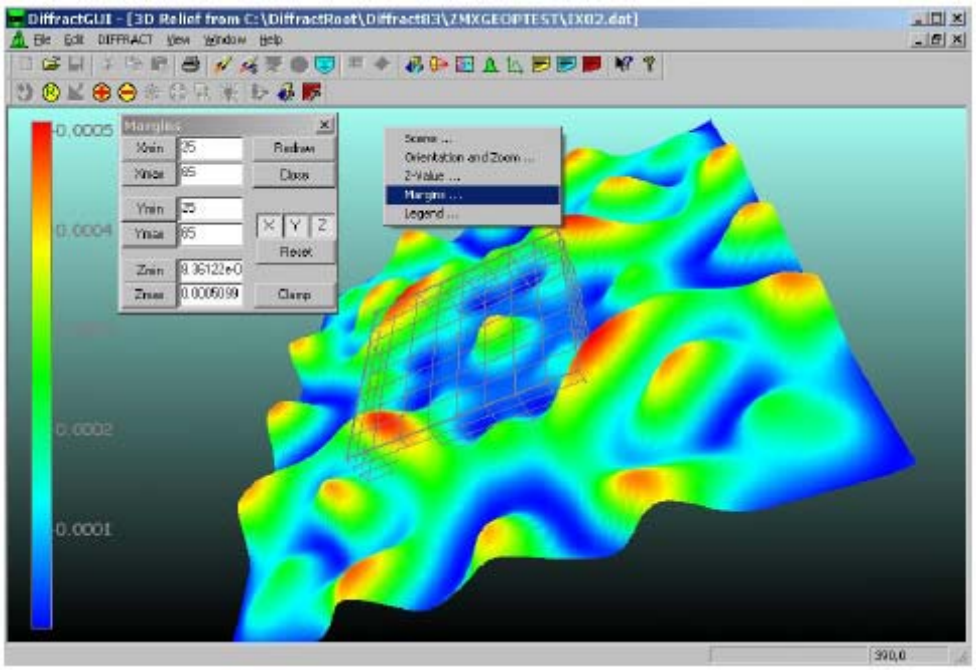


ReliefView に特有の Margin dialog box は右マウスクリックで現れる popup メニューの **Margin...**コマンドで起動する(次の図を見よ)。この dialog で制限しているグリッドの x-y-z 広がり　　は、Xmin, Xmax, Ymin, Ymax, Zmin および Zmax の入力によって決められる。Redraw ボタンを押すと ReliefView に入力した値を取り入れ、画像を書き直すように働く。もし、Clamp オプションをチェックすると、表面はグリッドで切断され、そうでなければ grid box の境界を越えても全体が描き直される。

最初の広がりを取り戻すには、下の図に示すように、小さい X, Y, Z チェックボタンの上にある Reset ボタンを使うと良い。この操作はチェックボタンをチェックした広がりだけに適用される。換言すれば、もし X, Y, Z ボタンをどれもチェックしなかった場合、Reset は動作しない。

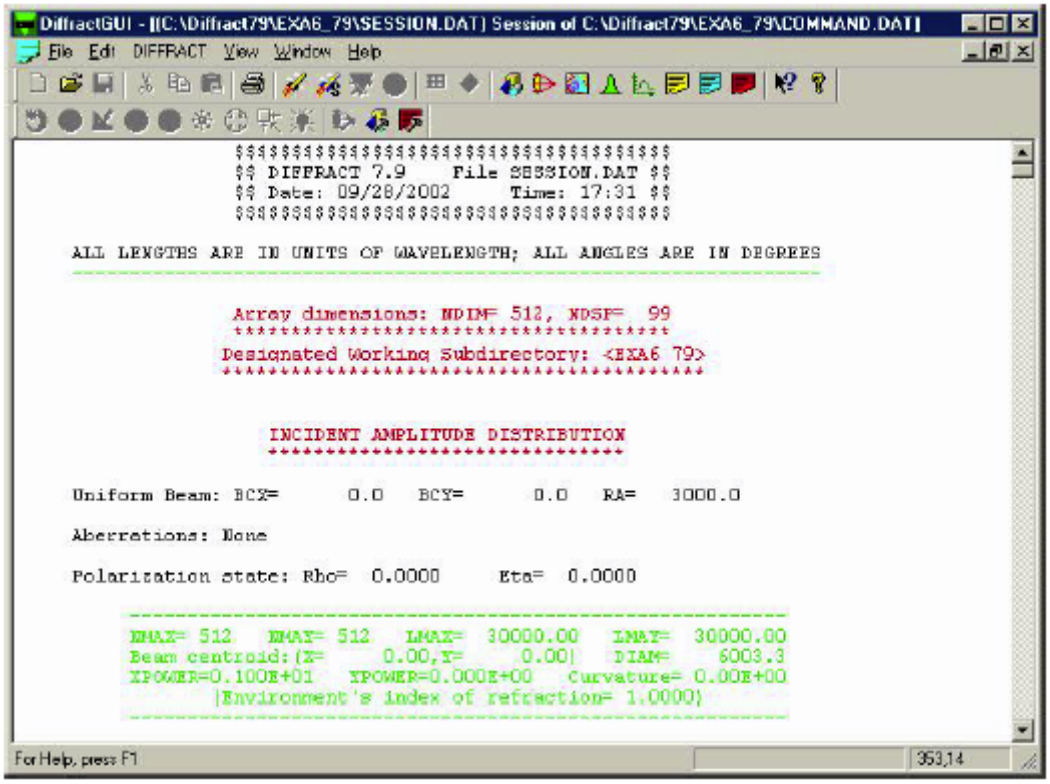


次の図は上と同じシーンを示したものであるが、Clamp オプションを作用しないようにしてある。



11. Session View

Session View は DIFFRACT の出力ファイル SESSION.DAT を早く見るのに便利である。

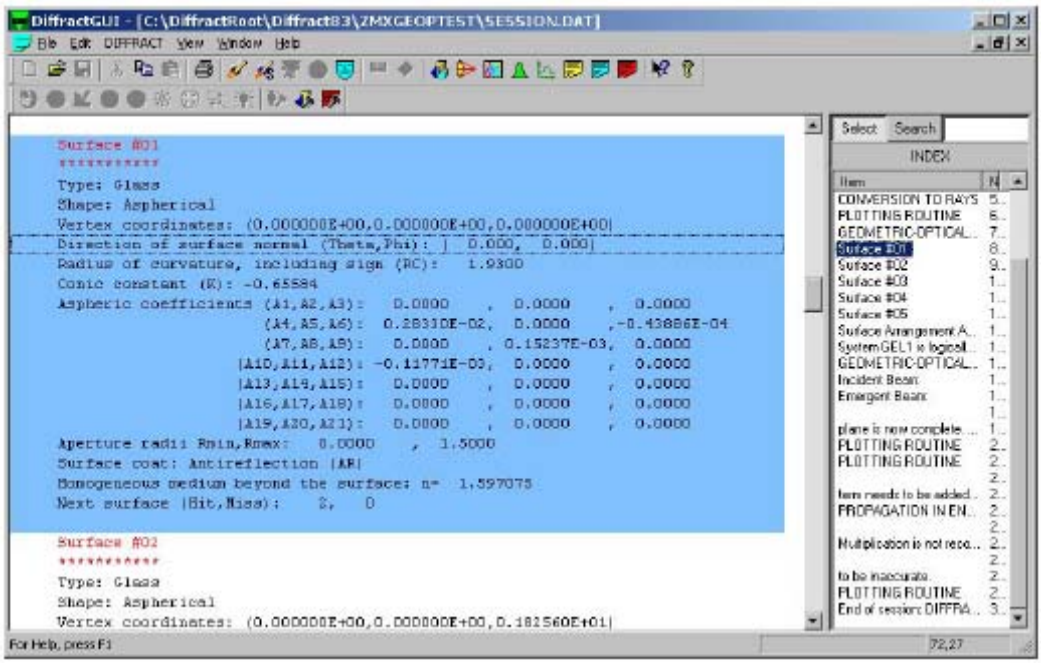


起動すると **Session View** は現在使っている COMMAND.DAT のディレクトリにある SESSION.DAT と名づけられたファイルの内容を表示する（もしファイルがないときは、ディスプレイは空のままである。）。Menu の **File-Open** から他のテキストファイル（たとえば SESSION.001）を読み込んで、表示することができる。テキストファイルは **Session View** で読み込み、表示できるが、Save option は存在しない。

次の規定によってテキストの色が決まっている。

- “*”と含むラインとその直前のラインは赤で表示される。
- “-“を含むラインは緑色で表示される。この色は”-“を含む別のラインか空のラインが現れるまで続く。

もし **Session View** が増大する SESSION.DAT を含む（すなわち、DIFFRACT が演算中）とき、ディスプレイは周期的に更新される。

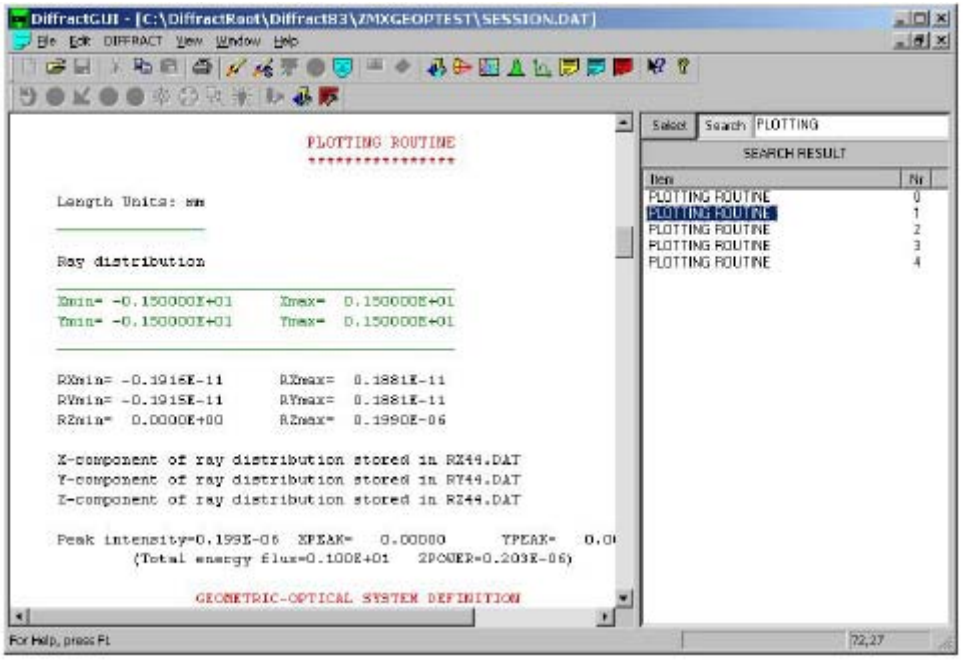


ScriptView と同様に、SessionView は画面の右側にリストと 2 個の小さなボタン（Select と Search）を持つ小さなコントロールパネルを持っている。リストは 2 つのオペレーションモード、index と search、を持っていて、これらは **Search** ボタンの状態でコントロールされている。Search ボタンがチェックされていない（上図）ときは、リストはインデックスモードで、すべてのラインはアステリスク（*********）でアンダーラインされている。サーチモードは次の図に示してある。

どちらのモードでも、どれかのリストが選択されると、SessionView は対応するテキストブロックを表示する。上図で見られるように、Session.dat ファイル内で、リスト入力が、関連あるアウトプットメッセージブロックの先頭（先頭と終わりがアステリスクラインで囲まれている）に置かれている空のラインを指示することに注意されたい。、たとえリスト入力がそれらの行にテキストを含まない場合でも、選択すれば session.dat ファイルの対応する場所にジャンプし、正しく動作する。

SessionView では、テキスト行はマウス左クリックで選択（あるいは非選択）できる。キーボードの Ctrl および Shift キーと関連して、選択リストに他の行を加えたり、取り除いたり出来る。そのやり方は他の Windows applications（たとえば標準の Windows Explorer プログラム）と同じである。SessionView は通常のテキストエディターではないので、選択はフルラインに限られ、すなわち行の 1 部分は選択できない。選択した行は標準の Edit-Copy コマンド（あるいは Ctrl-C）でクリップボードにテキストとしてコピーされ、他のテキストエディターに貼り付けることが出来る。

便利のために、上の図に示すように、Select オプションが備えられている。このオプションを可能にすると、index リストでクリックすれば対応するテキストブロックが自動的に選択され、既に存在する（すなわち前に選択した）ブロックに前の選択を削除しないでも追加することが出来る.。



Search mode に入るには、search edit box に希望のテキストをタイプし、その後で Search ボタンをクリックする。このクリックで、リストはそのテキストを含むすべての行を表示する。Search を洗練させるためには、Search ボタンを選択せずに、再度クリックする。

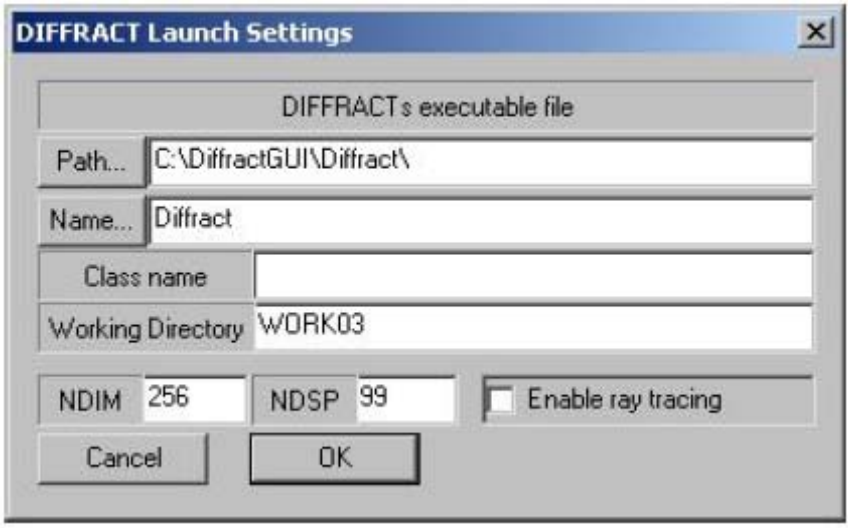
さらに便利のために、リストアイテムを左ダブルクリックまたは右クリックすると、コピーされ、リストディレクトリから search edit box に移すことが出来るが、リストが index mode のときのみである。さらに、右クリックはリストを search mode に変換し、search を初期化する。

12. DIFFRACT の起動

DIFFRACT を起動する前に、document を **COMMAND.DAT** という名前で希望する作業ディレクトリに保存しなければならない。DIFFRACT は Tool bar menu の **DIFFRACT_Launch(ZOOM)**あるいは **DIFFRACT_Launch(STEP)**でスタートすることができる。STEP モードは COMMAND.DAT の構築中に行うとよく、simulation parameter が有効かどうかをチェックすることができる。（ZOOM と STEP モードについての詳しい情報は、

DIFFRACT UserGuide を参照されたい)。作業ディレクトリの名前、NDIM、NDSP および光線追跡 option については、command line を通じて、直接 DIFFRACT にアクセスすればよい。これらのパラメーターは **DIFFRACT Launch settings** によって確認、訂正を行うことができる。

12.1 DIFFRACT 起動のための option (Options for DIFFRACT launcher)



DIFFRACT-Settings は dialog box を開く。Path はバックスラッシュ（日本語キーボードでは¥）で終えなければならないし、ファイル名は拡張子.exe をつけないで決定しなければならない。間違えるとエラーメッセージが出る。

ClassName は、DIFFRACT の main window を確認するパラメーターである。通常、ウインドウの表題テキストがウインドウを確認するために用いられるが、時々これでは不十分なことがある。たとえば、もし explorer がバックグラウンドで開かれ、偶然そのタイトルが”Diffract”であったとすれば、*DiffractGUI* が間違って explorer のウインドウを代わりに activate してしまう。このトラブルを避けるひとつの方法は、DIFFRACT の main window の Class Name を含めるとよい。Class Name は DIFFRACT77FRAME のようになるが、operating system と実行可能なファイルに与えた名前によって変更できる。要約すれば、ClassName は上述のような”explorer dummy launch”の時だけに必要であって、そうでなければ空欄にしておくべきである。もし、ClassName が空欄でない場合は、DIFFRACT の main window の正しい ClassName でなければならず、そうでなければ *DiffractGUI* は DIFFRACT の main window を見つけることができない。

DIFFRACT を起動できる path とファイル名、作業ディレクトリの名前、NDIM と NDSP の値、Ray_Tracing の可否が active document（すなわち COMMAND.DAT）が *DiffractGUI*

によって保存される（または読み込まれる）とき、次に述べるルールに従って、自動的に設定される。

- 作業ディレクトリーはそのファイルへの path から取り出される。たとえば、active document が **C:¥diffractGUI¥Diffract¥Work¥COMMAND.DAT** とすると、Launcher は **Work** を作業ディレクトリーと判断する。それにもかかわらず、プログラムが有効な path を document address から取り出すことができない時、作業ディレクトリーの path がレジストリに入力される。

- 起動可能な DIFFRACT の Path はまた、loaded/saved active document（すなわち active な COMMAND.DAT ファイル）の fully qualified path から取り出される。上の例では、取り出された path は **C:¥DiffractGUI¥Diffract¥**である。*DiffractGUI* はここで DIFFRACT の実行可能なファイルを見つけることができると期待される。

- NDIM パラメーター値は、すべての command block を検査し、すべての NMAX と NMAY パラメーターを比較して見出した最大値が DIFFRACT_Launch_Settings における NDIM の値を決定する。

- Enable ray tracing box は RAYS または FDTD_Export の command block が loaded/saved COMMAND.DAT の中に存在した時のみチェックされる。

NDSP パラメーターは *DiffractGUI* では変更されず、ユーザーの裁量に委ねられる。

ほとんどの場合、上述の parameter の選択が **DIFFRACT Launch settings** に適正となる。しかし、*DiffractGUI* の決定を覆し、別のパラメーターを設定するときは、DIFFRACT **Launch_Settings** 対話ボックスでパラメーターを修正すればよい。そうして対話ボックスを閉じ、直ちに DIFFRACT を起動する。すなわちファイルの保存・読み込みを行うまえに起動する。(COMMAND,DAT ファイルを保存または読み込みを行えば、起動パラメーターは上書きされる。したがって、dialog box を閉じてから Diffract を起動するまでの間、ファイルの保存・読み込みを一切行わないことが重要である。)

註

- DIFFRACT を対話モードで起動してはいけない。このモードでは、COMMAND,DAT ファイルの名前を変えてしまう。

- DIFFRACT の起動可能なファイルを含む（すなわち DIFFRACT.EXE が存在する）サブディレクトリに active な COMMAND.DAT を保存することを忘れてはいけない。次の起動のとき *DiffractGUI* は自動的にこのファイルを DIFFRACT に渡す。

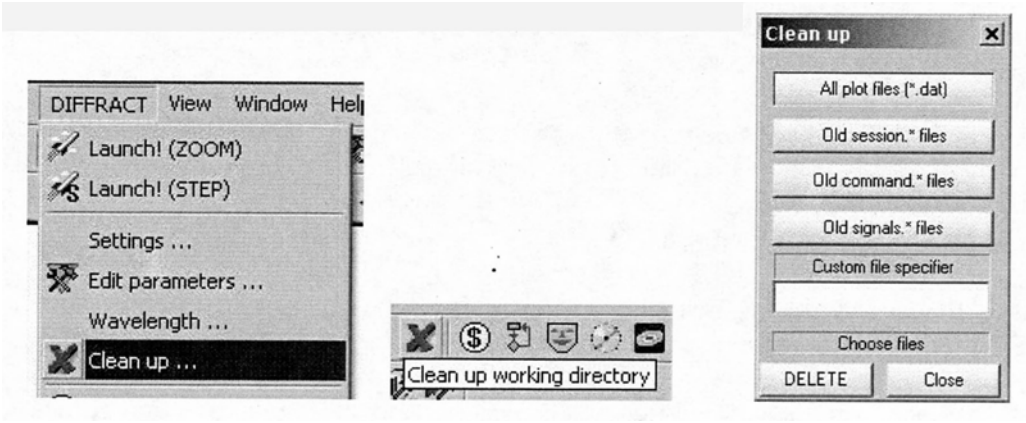
- Run を終わるたびに DIFFRCT を close されたい。そうでなければ、メモリで問題を起こす。

- DIFFRACT が計算を行いながら、*DiffractGUI* を使ってもよい。しかし、DIFFRACT で使用中の COMMAND.DAT ファイルを使ってはいけない。他のファイルは修正してもよいが、使用中の COMMAND.DAT を修正すると予期しない結果が起こる。

13. Working Directory の除去（Cleaning Up Working Directory）

DIFFRACT の実行中、あるシミュレーションの演算が実行され、それぞれの演算はいくつかの出力ファイルを作り出す。そのうちのあるものは演算に伴って上書きされ、他のものは新しく作られるが、コピーされて記録のために残されるものもある。

そのようなファイルの例外的な集積を避けるために、それらを削除するために CleanUp 命令が使われる。



Clean up・・・命令をクリックすると、Clean up dialog box が現れ、次に示したグループのファイルを 1 つあるいはそれ以上を削除することが出来る。

All Plot files: ?n??.dat, ?n??.bmp あるいは?n??.tif などの全ファイル、ここに?は文字あるいは数字、n は X, Y, Z, S. L, T または P の文字。

Old session files: session.???に該当するすべてのファイル。?は文字または数字。安全のために、session.dat は除外し、削除してはいけない。

Old command files: command.???に該当するすべてのファイル。?は文字または数字。安全のため、command.dat は除外し、削除してはいけない。

Old signal files: signal.???に該当するすべてのファイル。?は文字または数字。安全のため、signals.dat は除外し、削除してはいけない。

Custom file specifier: edit box における決められたフィルターにマッチするすべてのファイル。Wildcard 文字 ”?” および “*” がフィルターを特定するのに用いられる。たとえば、“*.txt”は拡張子 txt を持つすべてのファイルにマッチする。”test.???”は 3 個までの任意の文字を拡張子、または拡張子を持たないすべてのファイルを削除する。

注意

削除したファイルは復元できない。特に、custom-file フィルターoption は、他のプログラムによって作業ディレクトリの中に作られる valuable file をも削除することを可能とするため、この特徴は留意して使わなければならない。たとえば、”*.*” または “*” のフィルターは作業ディレクトリからすべてのファイルを削除することがある。この場合、削除をキャンセル機会を与える警告が表示される。

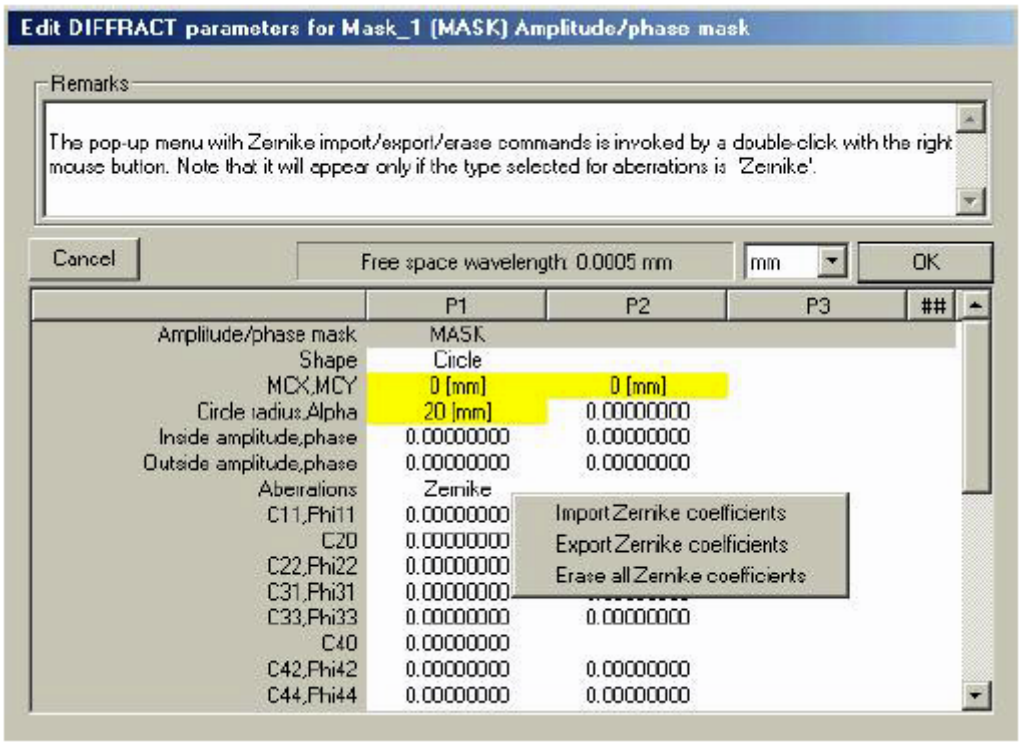
他の方法として、command.dat, session.dat および signals.dat のファイルは CleanUP によって削除されることは無い。他のすべてのファイルは、もし削除が失敗したとき（何らかの理由によって、たとえばもしあるファイルが read-only あるいは他のプログラムでロックされたとき）そのファイルを削除することを再び企てることは出来ないので、削除を残すことが出来る。

すべての削除の処置を **LogView** で追跡することができ、CleanUP が終わったことを検査することができる。

14. Zernike 収差係数の Import/Export（Import/Export of Zernike Coefficients）

DiffractGUI は Zernike の収差係数を、それらが定義されたところで、Import（読み込み）または Export（ファイルとして取り出し）できる。これは"*Edit DIFFRACT parameter*" dialog box または*$-command* editor box で行うことができる。ユーザーは object parameter のリストから収差のタイプ"Zernike"を選び、リスト内部のどこかでマウスの右ボタンをダブルクリックする。Pop-up menu が次の command option と共に現れる。

- **Import Zernike Coefficients** ― 外部ファイルから係数の新しい数値を読み込む。
- **Export Zernike coefficients** ― 外部ファイルへ係数の現在の値を書き込む。
- **Erase all Zernike coefficients** ― すべての係数をゼロにセットする。



14.1 ファイル形式（The File Format）

Zernike 多項式W_{nm}は cosine のみ、あるいは sine-cosine 混合項として表すことができる。下の式(1a)および(1b)を参照されたい。第１の場合（タイプ１）は *DIFFRACT* で使われ、ユーザーはそれぞれの多項式の係数C_{nm}とΦ_{nm}を決めなければならない。第２のケース（タイプ２）は係数A_{nm}（cosine 項）とB_{nm}（sine 項）を決めなければならない。

$$W_{nm} = C_{nm} \cdot f(r) \cdot \cos(m(\Phi - \Phi_{nm})) \qquad (1a)$$

$$W_{nm} = A_{nm} \cdot f(r) \cdot \cos(m \cdot \Phi) + B_{nm} \cdot f(r) \cdot \sin(m \cdot \Phi)) \qquad (1b)$$

DiffractGUI は両者のデータ形式を読み込むことができる。タイプ２の係数（A_{nm}, B_{nm}）を読み込んだとき、*DiffractGUI* は対応する C_{nm}, Φ_{nm} を次のように計算する。

$$\Phi_{nm} = \arctan\left(\frac{B_{nm}}{A_{nm}}\right) \cdot \frac{1}{m}$$
$$C_{nm} = \sqrt{A_{nm}{}^2 + B_{nm}{}^2} \qquad (2)$$

ファイルの各行は１係数のみを含んでいる。このラインは半分ずつ２個に分割され、#で区切られている。空のスペースは数えない。もし#が無いとき、もし１個以上の#があるとき、または１個または両方の半ラインが空のときは、そのラインはスキップされ、インタープリーターによって無視される。このルールにより、#で始まるラインはコメントラインとみなされる。

ラインの最初の半分は、空のスペースで分離された、１つまたはそれ以上の数を含むが、最後の数のみが係数の値とみなされる。ラインの残り半分はそのラインに含まれる係数の種類を特定し、#の右に続くキーワードを見ればよい。もしキーワードが C_{nm} または PHI_{nm}（n および m は 0, 1, 2, 3, 4, 5, ...を表す）の形を持っているときは、Zernike の多項式のタイプ１の表示であることが分かり、係数 C_{nm} または Φ_{nm} がラインから直接得られる。他方、キーワードが A_{nm}, B_{nm} または $f(r)\cos(m\Phi)$, $f(r)\sin(m\Phi)$ という直接的な表現の場合はタイプ２と考え、C_{nm}, Φ_{nm} を式(2)によって計算すればよい。

註

- Zernike ファイル形式は DIFFRACT 7.7, 7.9, 8.1, 8.2 および 8.3 で共通である。したがって上記の Verson 間では共通で使用できる。
- インタープリーターは大文字・小文字を区別する。したがってキーワードは下記の３つの例（太字のテキスト）に示すように正確に表示しなければならない。
- キーワードは#記号の直後に置かなければならない。これらは任意のコメントの後ろに置いても良い（例を見よ）。
- 係数の順序は任意である（下記の例２および３を参照のこと）。
- タイプ２のパラメーターを含むファイルでは、A_{nm} は通常対応する B_{nm} を伴い、逆もまた同様である（隣接するラインに置かなくても良い）。このルールの例外は係数 A20, A40, A60 および A80 である。もし、ある A_{nm} が対応する B_{nm} を伴っていない（逆も同様）ときは、係数 B_{nm} はゼロと置かれる。

- Zernike の係数ファイルは 1 個のファイルで完結していないこともある。その場合は、ファイルに書かれていない係数はゼロと置かれる。
- *DiffractGUI* は常にタイプ 1 形式で係数を書き出す（Export）、すなわち C_{nm} , PHI_{nm} である。

First type(タイプ１)の Zernike 係数を含む典型的な **Import_Files** を示す。*DiffractGUI* はリストされたキーワードのみを識別する。たとえば、C74 のような受け入れられない係数があると *DiffractGUI* にそのラインを読み飛ばす。インタープリーターは各ラインの先頭にある **index number** を無視する。次のファイルの形式は Zernike 係数をファイルに **export**(書き出す)するときにも用いられる。

```
# (DIFFRACT) Export from C:\Diffract81\MaskTest.dat
# Zernike coefficients (Mask_1)
# Index  Value    Coefficient
  01 -0.116068000000        # C11
  02 -13.2968000000         # PHI11
  03 -0.353276000000        # C20
  04  0.814565000000        # C22
  05 -35.2386000000         # PHI22
  06  0.105213000000        # C31
  07 -21.7444000000         # PHI31
  08  0.0870823000000       # C33
  09  19.9486000000         # PHI33
  10 -0.00279300000000      # C40
  11  0.0758357000000       # C42
  12  34.1537000000         # PHI42
  13  0.0443355000000       # C44
  14 -21.3194000000         # PHI44
  15  0.0108659000000       # C51
  16  3.80460000000         # PHI51
  17 -0.00144002000000      # C53
  18  29.9072000000         # PHI53
  19 -0.0100861000000       # C55
  20  8.86905000000         # PHI55
  21 -0.000188000000000     # C60
  22  0.00332505000000      # C62
  23  8.69706000000         # PHI62
  24 -0.00887984000000      # C64
  25 -17.8045000000         # PHI64
  26 -0.00948549000000      # C66
  27  14.4036000000         # PHI66
  28  0.0141526000000       # C71
  29  33.8046000000         # PHI71
  30  0.0140322000000       # C73
  31 -21.6941000000         # PHI73
  32  0.00613673000000      # C75
  33  9.99418000000         # PHI75
  34  0.000235332000000     # C77
  35  11.9860000000         # PHI77
  36 -0.00716300000000      # C80
  37  0.00591692000000      # C82
  38 -15.9344000000         # PHI82
  39  0.000000000000        # C84
  40  0.000000000000        # PHI84
  41  0.000000000000        # C86
  42  0.000000000000        # PHI86
  43  0.000000000000        # C88
  44  0.000000000000        # PHI88
```

Second type(タイプ 2)の Zernike 係数を含む典型的な **Import_File** を示す。このファイルに含まれたボールド文字キーワードは完全で、現在 *DiffractGUI* で認識されるすべてを含んである。

```
#Index Coefs(microns) Equation
   1    -0.112956  # rcos(t) (X Tilt)
   2     0.026695  # rsin(t) (Y Tilt)
   3    -0.353276  # 2r^2-1 (Focus)
   4     0.272212  # r^2cos(2t) (0 Astigmatism)
   5    -0.767735  # r^2sin(2t) (45 Astigmatism)
   6     0.097727  # (3r^2-2)rcos(t) (X Coma)
   7    -0.038978  # (3r^2-2)rsin(t) (Y Coma)
   8    -0.002793  # 6r^4-6r^2+1 (Spherical)
   9     0.043744  # r^3cos(3t)
  10     0.075298  # r^3sin(3t)
  11     0.028031  # (4r^2-3)r^2cos(2t)
  12     0.070465  # (4r^2-3)r^2sin(2t)
  13     0.010842  # (10r^4-12r^2+3)rcos(t)
  14     0.000721  # (10r^4-12r^2+3)rsin(t)
  15    -0.000188  # 20r^6-30r^4+12r^2-1
  16     0.003650  # r^4cos(4t)
  17    -0.044185  # r^4sin(4t)
  18    -0.000007  # (5r^2-4)r^3cos(3t)
  19    -0.001440  # (5r^2-4)r^3sin(3t)
  20     0.003173  # (15r^4-20r^2+6)r^2cos(2t)
  21     0.000994  # (15r^4-20r^2+6)r^2sin(2t)
  22     0.011760  # (35r^6-60r^4+30r^2-4)rcos(t)
  23     0.007874  # (35r^6-60r^4+30r^2-4)rsin(t)
  24    -0.007163  # 70r^8-140r^6+90r^4-20r^2+1
  25    -0.007213  # r^5cos(5t)
  26    -0.007050  # r^5sin(5t)
  27    -0.002859  # (6r^2-5)r^4cos(4t)
  28     0.008407  # (6r^2-5)r^4sin(4t)
  29     0.005912  # (21r^4-30r^2+10)r^3cos(3t)
  30    -0.012726  # (21r^4-30r^2+10)r^3sin(3t)
  31     0.005025  # (56r^6-105r^4+60r^2-10)r^2cos(2t)
  32    -0.003124  # (56r^6-105r^4+60r^2-10)r^2sin(2t)
  33    -0.000592  # r^6cos(6t)
  34    -0.009467  # r^6sin(6t)
  35     0.003947  # (7r^2-6)r^5cos(5t)
  36     0.004699  # (7r^2-6)r^5sin(5t)
  37     0.000025  # r^7cos(7t)
  38     0.000234  # r^7sin(7t)
  39      0        # (28r^4-42r^2+15)r^4cos(4t)
  40      0        # (28r^4-42r^2+15)r^4sin(4t)
  41      0        # (8r^2-7)r^6cos(6t)
  42      0        # (8r^2-7)r^6sin(6t)
  43      0        # r^8cos(8t)
  44      0        # r^8sin(8t)
```

上記の（長い）キーワードの代わりに、*DiffractGUI* は先のリストと１対１に対応する次のような（非常に短い）別名のキーワードを認識する。

```
#Index Coefs(microns) Equation
   1    -0.112956  # A11 (X Tilt)
   2     0.026695  # B11 (Y Tilt)
   3    -0.353276  # A20 (Focus)
   4     0.272212  # A22 (0 Astigmatism)
   5    -0.767735  # B22 (45 Astigmatism)
   6     0.097727  # A31 (X Coma)
   7    -0.038978  # B31 (Y Coma)
   8    -0.002793  # A40 (Spherical)
   9     0.043744  # A33
  10     0.075298  # B33
  11     0.028031  # A42
  12     0.070465  # B42
  13     0.010842  # A51
  14     0.000721  # B51
  15    -0.000188  # A60
  16     0.003650  # A44
  17    -0.044185  # B44
  18    -0.000007  # A53
  19    -0.001440  # B53
  20     0.003173  # A62
  21     0.000994  # B62
  22     0.011760  # A71
  23     0.007874  # B71
  24    -0.007163  # A80
  25    -0.007213  # A55
  26    -0.007050  # B55
  27    -0.002859  # A64
  28     0.008407  # B64
  29     0.005912  # A73
  30    -0.012726  # B73
  31     0.005025  # A82
  32    -0.003124  # B82
  33    -0.000592  # A66
  34    -0.009467  # B66
  35     0.003947  # A75
  36     0.004699  # B75
  37     0.000025  # A77
  38     0.000234  # B77
  39      0         # A84
  40      0         # B84
  41      0         # A86
  42      0         # B86
  43      0         # A88
  44      0         # B88
```

14.2　A_{nm} と B_{nm} の単位（Units for A_{nm} and B_{nm}）

次の測定単位が A_{nm} と B_{nm} 係数に適用できる。
[cm] [mm] [um] [nm] centimeter millimeter micrometer nanometer micron

Zernike の Import ファイルの各ライン（コメントラインを含む）は、最初に上記のリストにある単位を示すキーワードの存在がチェックされる。もし、これらの一つが存在したときは、その単位はそのラインと、それと違った単位を表示したラインが現れるまで、それに続くすべてのラインの単位となると考えられる。その単位を使って、すべての A_{nm}, B_{nm} および C_{nm} は、式(2)を適用する前に、*DiffractGUI* で認識されている真空中の波長で基準化されている。

与えられたラインでは、単位のキーワードは#記号の前には現れない。このキーワードは連続した綴り（ブランクは不可、white space"_"は可能）でなければならず、略号はブラケット"[ ]"で囲まなければならない。単位を表すキーワードはどの位置においても良く、大文字でも小文字でも良い。たとえば、次のラインでは **micron** が認識される。

```
# somecommentMicRonsnm another comment
```

nm とか mm のような部分綴りは認識されない。これらは"[“と"]"で囲まれていないからである。次のラインは有効なキーワードがない。これは **micron** という字が連続していないからである。

```
# somecommentM Ｉ c Ronsnm another comment
```

１つのラインに多数の単位キーワードを表示してはいけない。

註：　*DiffractGUI* は常に Zernike 係数を micron 単位で export する。

15. 位相/振幅 Object として使われる Mask Data Files（Mask Data-files used as Phase/Amplitude Objects）

DIFFRACT の Extended Phase/Amplitude object(XOBJ)機能は、外部データファイル（ASCII またはバイナリー）を用いて光学系の中に位相/振幅マスクを挿入することができる。これらの外部データファイルは通常前もって作成され、*.DAT という名前でハードディスクの DIFFRACT の作業ディレクトリに保存されている。XOBJ 機能はそのようなデータファイルを作る option（Create モードで）を持っているが、XOBJ に要求される形式であれば、どのようなソフトウエアを使って作成しても良い。

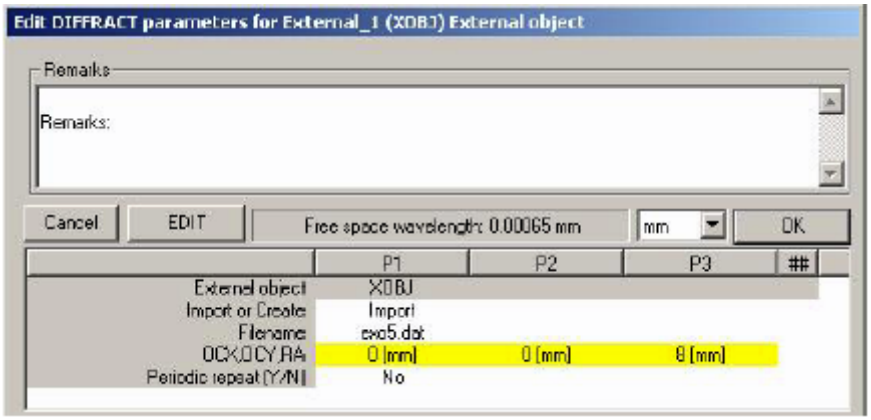
DiffractGUI の External_Mask editor は外部データファイル（ASCII またはバイナリー）を作ることができる新機能であり、XOBJ 機能により後で DIFFRACT に読み込むことができる。この Mask_Editor は DIFFRACT で作られたビームの断面(*.DST ファイルとして保存)を出発点（初期マスク）として読み込むことができ、その上にマスクのデータファイルを構築する。同様に、Mask_Editor は、前に作られて XOBJ に順応する*.DAT（ASCII またはバイナリー）として保存されているマスクデータファイルを読み込んで作ることができる。そのようなマスクデータファイルは Mask_Editor で修正され、新しい XOBJ に適合するデータファイルとして保存される。他の方法として、ユーザーは希望するマスクの位相/振幅パターンを Mask_Editor が受け入れ可能な数式で決めることもでき、そのマスクを XOBJ 適合のデータファイルとしてハードディスクに保存しても良い。初期マスクの位相/振幅パターンファイルが*.DST または*.DAT、あるいはユーザーが決めた数式によるものであったかにかかわらず、Mask_Editor に要求される最終結果は常に同一である。すなわち XOBJ 機能によって後で DIFFRACT に読み込むことができる XOBJ 適合の*.DAT ファイルを作ることである。

DiffractGUI の External_Mask editor は次のどれかの方法で起動することができる。

- Toolbar の mask-icon による方法。

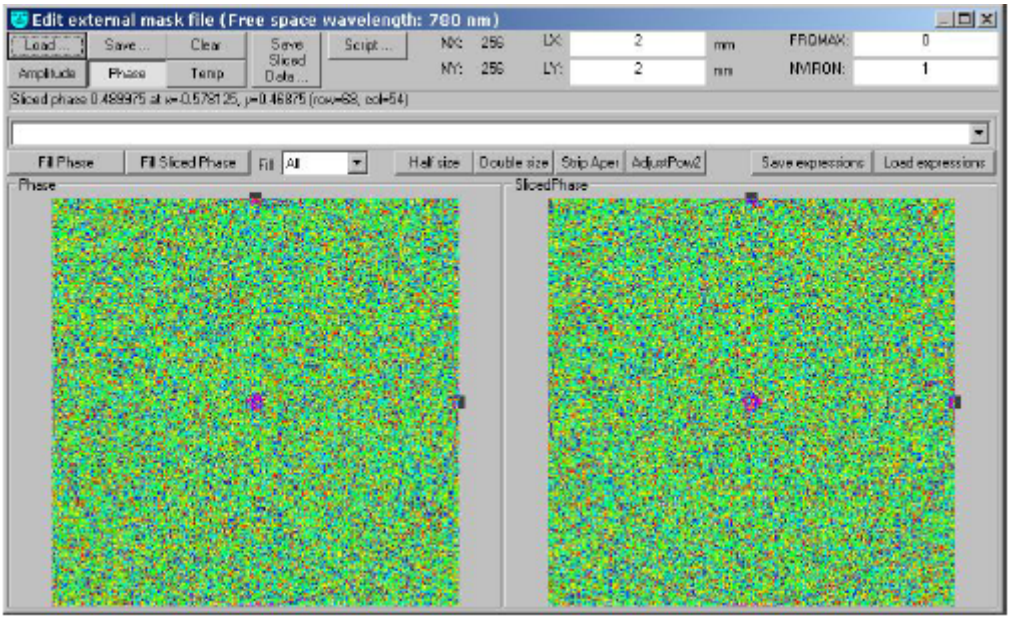
- Menu command DIFFRACT – Edit external mask による方法。

- XOBJ を編集するとき **Edit DIFFRACT Parameters** dialog box の中で、**EDIT** ボタンを押すか、またはパラメーターリスト領域でマウスの右クリックを使う方法。



Mask_Editor Dialog Box

他のウインドウと同様に、Mask_Editor dialog box（下に示す）はサイズが変更でき、スクリーン全体にまで拡張することができる。



註：Mask_Editor は *DiffractGUI* を使って計算する前に閉じなければならない。

Mask_Editor は次の 6 つの内部マトリックスを取り扱う。それぞれは NX 個の列と NY 個の行を持っている。

- Amplitude（振幅）
- Sliced Amplitude（スライス*された振幅）
- Phase（位相）
- Sliced Phase（スライスされた位相）
- Temp1
- Temp2

これらは **Amplitude, Phase および Temps** ボタンを使って一度に一対が表示される。表示された一対は（Amplitude, Sliced Amplitude）,（Phase, Sliced Phase）および（Temp1, Temp2）である。Default pair（Phase: Sliced Phase）は常に起動画面に表示される。Amplitude(A), Phase(Φ), 実数（Re)、虚数（Im）成分は sliced distribution と同様に

$$A = \sqrt{\mathrm{Re}^2 + \mathrm{Im}^2}, \qquad \Phi = \arctan(\mathrm{Im}/\mathrm{Re})$$
$$\mathrm{Re} = A\cos(\Phi), \qquad \mathrm{Im} = A\sin(\Phi) \tag{1}$$

Temp1 と Temp2 のマップは script を実行する間の中間的データ記憶、あるいは bitmap またはカスタムファイルの読み込みにおけるデータの初期抑制のためのものである。Temp1 および Temp2 は maskEditor の上部にある Temp ボタンを押せば見ることが出来る。Temp1, Temp2 の場合不可能な slicing を除き、他の図表作成のときはいつでも作動できる。

6 つの matrix すべては初め random number（0 から 1 までの領域）で満たされている。これらのマトリックスにデータを入力するには、いくつかの異なった方式がある。

- 振幅マトリックスを 1 の成分、位相マトリックスを 0 の成分で満たすには **Clear** ボタンを用いる。こうすれば 2 つの初期マトリックス（Primary matrix）はその sliced matrix にコピーされる。

- 既に保存されている分布ファイル(*.DAT)からデータを振幅および位相マトリックスに読み込むには、**Load Beam** ボタンを用いる（*.DST ファイルについては、DIFFRACT の取り扱い説明書を参照されたい）。2 つの初期マトリックス（振幅と位相）はその sliced matrix にコピーされる。すべてのマトリックスは読み込まれたファイルのサイズ NX ×NY に変更される。

- 既に保存されている外部マスクファイル(*.DAT)からデータを振幅および位相マトリックスに読み込むには、**Load Mask** ボタンを用いる（外部位相/振幅マスクについては、

* slice（スライスする）：連続した分布をいくつかのレベルで段階的に表すことを言う。

DIFFRACT の取り扱い説明書を参照されたい)。このようにして初期マトリックスはその slice matrix にコピーされる。すべてのマトリックスは読み込まれたファイルのサイズ NX×NY に変更される。

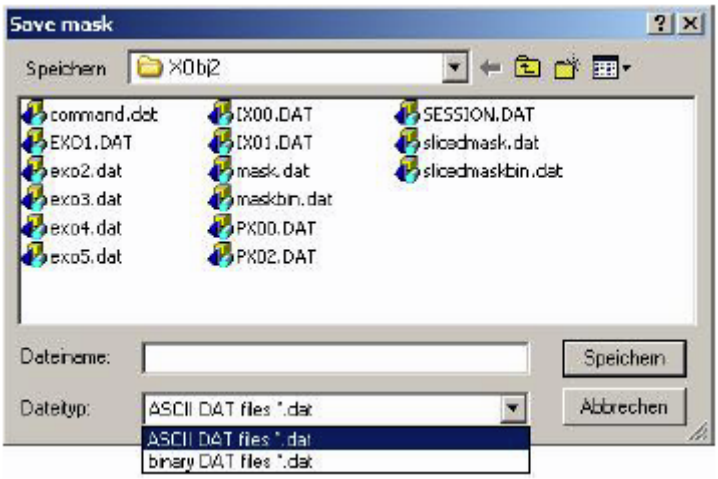
- 任意の数式（**Fill** ボタンの上の expression-box にタイプする必要がある）で表現されるマトリックスを作るためには、**Fill** ボタンの一つを用いる。ユーザーが定義した数式によって計算されたデータは、ディスプレイされた最終マトリックスに挿入される。Fill の機能については、後でこの付録で述べる。

マトリックスのサイズ(NX, NY)は **Double size** または **Half size** ボタンで 2 の倍数で増減する（2 以外の倍数は使えない)。マトリックスサイズの増大は、現存するデータの周りに 0 を追加するが、中心は新しい拡大されたマトリックスの中心に置く。マトリックスサイズの縮小は現存するフレームを周辺から縮小するが、中心に近い部分は変えない。最大（最小）のマトリックスサイズは、NX または NY の大きい（小さい）方が 2048 (32) である。

注意：マトリックスサイズを縮小すると、外れたデータは消去される。

NX, NY の値だけでなく、Mask_Editor はパラメーターLX, LY および FRQMAX も表示する。これらは既に保存されている分布（*.DST または*.DAT ファイル）に伴うパラメーターである。**Double size** および **Half size** ボタンによるピクセル数 NX, NY の変更は、自動的にメッシュサイズ LX, LY を変更する。ユーザーはまた、LX, LY の値を対応した box にタイプして、Enter を押せば希望の値に変更できる。マスクの最高空間周波数の推定値（FRQMAX）も、[0, 1]の範囲の実数で、ユーザーが設定できる。

Save Mask ボタンをクリックすれば、位相と振幅マトリックスは DIFFRACT の XOBJ コマンドによる読み込みに適した外部データファイル(*.DAT)として保存することができる。このファイルは File Open dialog box を使って、下図に示すように ASCII またはバイナリーファイル形式で保存できる。**Save Sliced Mask** ボタンは、スライスされた振幅、位相マトリックスに対して、同様な作用をする。



2 つの初期マトリックス（振幅と位相）には種々の方法でデータの入力ができるのに対し、スライスされたマトリックスは初期マトリクスから slicing operation によってデータをコピーすることによって作られる。スライスされたマトリックスを作るには、Slice Control を開き、slicing level(20 まで)を設定して、**Slice!**ボタンをクリックする。Slice Control は **Amplitude** または **Phase** のうち、開きたい方のボタンをクリックすればよい。表示した画像上で右マウスクリックすれば、Pop-up menu が開き、表示に関する種々の option を変更することができる。Pop-up menu の **Slice…**によっても（**Amplitude** または **Phase** をクリックするのと同じように）Slice Control を開くことができる。Slice control では次のような option が可能となる。

- 新しい level-marker を挿入するには、着色したチャンネルバー内部を左マウスクリックすればよい。（Slice Control で 20 までの marker を作ることができる。）

- 既存の level-marker を消去するためには、その marker の上で右マウスクリックをすればよい。

- level-marker の位置を変えるには、それをマウスカーソルでドラッグすればよい。Marker は隣接する marker との間の領域内でのみ移動できる。

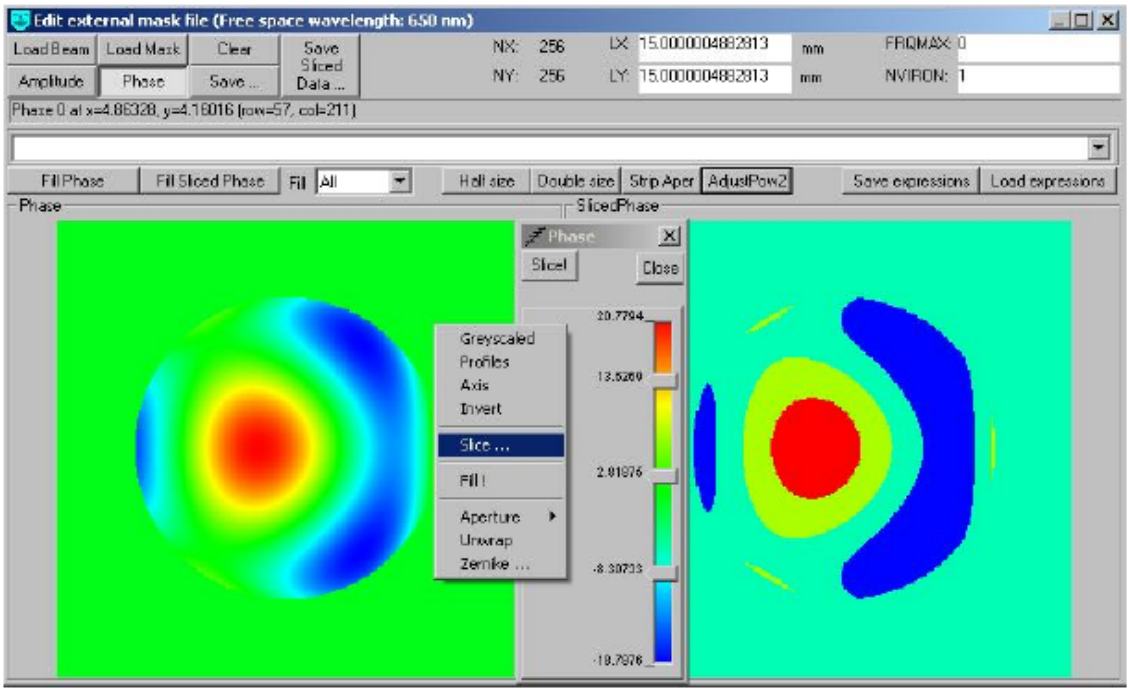
- Slice Control はサイズを変えることができる。したがって、level 数が大きいときはユーザーは Slice Control を伸ばして、見やすくすることができる。

- **Slice!** operation が起動しても、level-marker を定義しないときは、既存の初期マトリックス（振幅、位相）が Sliced matrix にそのまま直接コピーされる。

- Mask_Editor は(振幅と位相に対応して)２つの slice control を表示する。Mask_Editor が起動している間は、それぞれの Slice Control は、level marker を保存したまま、閉じたり再び開いたりすることができる。最初に開いたときは、それぞれの Slice control は level marker のない状態で表示される。

- Slice Control の最大・最小値は、Slice Control が開いているときのみ意味を持つ。したがって、マトリックスに他の処理を行うときは、Slice Contro を閉じた方がよい。

下図は Slice Control と pou-up menu が開いている Mask_Editor の例である。左側に表示されている（初期）位相マトリックスは、前に DIFFRACT によって作られ（また保存され）た分布ファイルから読み込まれたものである。右側のスライスされた位相の表示は Slice Control で指示された level で初期位相マトリックスをスライスした結果である。



数式と計算インタープリターによる読み込み　*DiffractGUI* の Mask Editor の強力な機能は **Fill** であって、Editor の **Fill** ボタンをクリックするか、または pop-up menu から **Fill!** をえらんで起動することができる。この機能は **Fill** ボタンの上の expression box に書いた計算式を計算インタープリターが受け入れる。インタープリターは数式を解釈し、目的のマトリックスの内容を計算結果と入れ替える。

インタープリターは基本的な算術記号 /, *, +, －を認識する。括弧()は算術の領域を定めるのに無制限の連鎖的に用いられる。さらに、次のような演算が認識されている。

- ｜（論理 OR）： a|b もしａとｂが 0 のとき 0 を返す、そうでなければ 1.
- & （論理 AND）： a & b ａとｂがゼロでなければ 1 を返す、そうでなければ 0
- = （比較）： a = b ａとｂが等しければ 1 を返す、そうでなければ 0
- ^（冪）： a ^ b ａのｂ乗を返す
- <（より小さく）： a < b ａがｂより小さければ 1 を返す、そうでなければ 0
- >（より大きく）： a > b ａがｂより大きければ 1 を返す、そうでなければ 0
- %（モヂュラー演算）： a % b a/b の剰余を返す、a, b が整数のときのみ有効。

変数*のない関数 **rand( )**は 0 と 1 の間でランダム数を発生する。

次の 1 変数関数は計算式に用いてもよい。: **sin, cos, tan, sinh, cosh, tanh, dsin, dcos, dtan, atan, asin, acos, exp, ln, log, sqrt, abs, not, sgn, floor, celi.** すべての三角関数では、変数の角度はラヂアンであるが、**dsin, dcos, dtan** は度で表す（その意味で関数名に d が付いている)。逆三角関数で得られる角度はラヂアンである。**ln** は自然対数、**log** は 10 を底とする対数、**not** は変数が 0 なら 1 を返しその他の場合は 0、**sgn** は符号判別で変数が 0 なら 0、変数が正なら 1、負なら 0 を返す。**floor** は変数より小さい最大の整数を返し、**ceil** は変数よりも大きい最小の整数を返す。:

次の 2 変数関数 **and, or, xor, mod, pow, max, min** も計算式に使うことができる。論理関数 **and, or, xor** はそれらの論理演算の結果 0 または 1 を返す。0 を返す論理演算は FALSE と解釈され、そうでなければ TRUE と考える。**mod** はモジュラー関数であって、最初の変数を 2 番目の変数で割り算してその剰余を返す。**%**演算とは違って、**mod** 関数は非整数・整数ともに同様に適用できる。**pow** は^演算と同じで、第 1 の変数の第 2 変数乗を返す（第 2 変数は整数でなくてもよい)。**max** および **min** はそれぞれ 2 つの変数の大きい方および小さい方を返す。

計算のインタープリーターはまた 3 つの変数を持つ関数 **cond**(a,b,c)を認識する。この関数は第 1 変変を調べ、もしそれがゼロでないならば関数は第 2 変数を返し、そうでないときは第 3 変数を返す。予め定義された変数（下を見よ）を使えば、この関数はマトリックスの限られた部分だけを変えるのに非常に有用である。

さらに、次に示すような一般に使用できる関数が備えられている。

atan(y,x): arctangent(y/x)を計算し、0 と 2π の間の値を返す。

---

* argument：引数ともいう

atan180(y,x): arctangent(y/x)を計算し、－πとπとの間に値を返す。

dist(x,y): $\sqrt{x^2+y^2}$ を計算する。

J0(x), J1(x), Jn(x)：第 1 種 Bessel 関数の 0 次、1 次、n 次を計算する。
Y0(x), Y1(x), Yn(x): 第 2 種 Bessel 関数の 0 次、1 次、n 次を計算する。
sinc(x): $\sin(x)/x$ を計算し、x=0 の場合 1 を返す。
Jinc(x): $J_1(x)/x$ を計算し、x=0 の場合 1/2 を返す。
factorial(n): n!を計算する。
gamma(x): ガンマ関数$\Gamma(x)$ を計算する。

erf(x):エラー関数$\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$ を計算する。

erfc(x): 補足エラー関数$\frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt$ を計算する。

beta(x,y): ベータ関数$\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$ を計算する。

bincoe(n,k): binormal coefficient$\binom{n}{k}=\frac{n!}{k!(n-k)!}$ を計算する。

インタープリターは**PI**を**π**(3.1415927…)と認識し、**EULER**を自然対数の底**e**(2.71…)、**true** を 1、**false** を 0 と認識する。また、次の定義された変数が（数字の代わりに）計算式に用いることができる。

- **I**(または **ROW**)はマトリックスの行を表し、**J**(または **COL**)は列を表す。
- **X, Y** は描画の x, y 座標を表す。
- **R, PHI** は描画の半径と方位角φを表す。**R** は中心からの距離である。**PHI** は半径が$x-$軸となす角度である。この角度は$x-$軸から反時計方向に測り、**PHI** の値[0：360°]は一般に度で表す。
- **V** はマトリックス成分の現在の値（数式の評価前）を表す。
- **VMAX, VMIN** はマトリックスの最大値、最小値を表す。
- **A** は振幅マトリックスの現在の値を表す。
- **P** は位相マトリックスの現在の値を表す。
- **SA** はスライスされた振幅マトリックスの現在の値を表す。
- **SP** はスライスされた位相マトリックスの現在の値を表す。

注意：Mask_Editor の計算インタープリターは大文字・小文字を区別する。

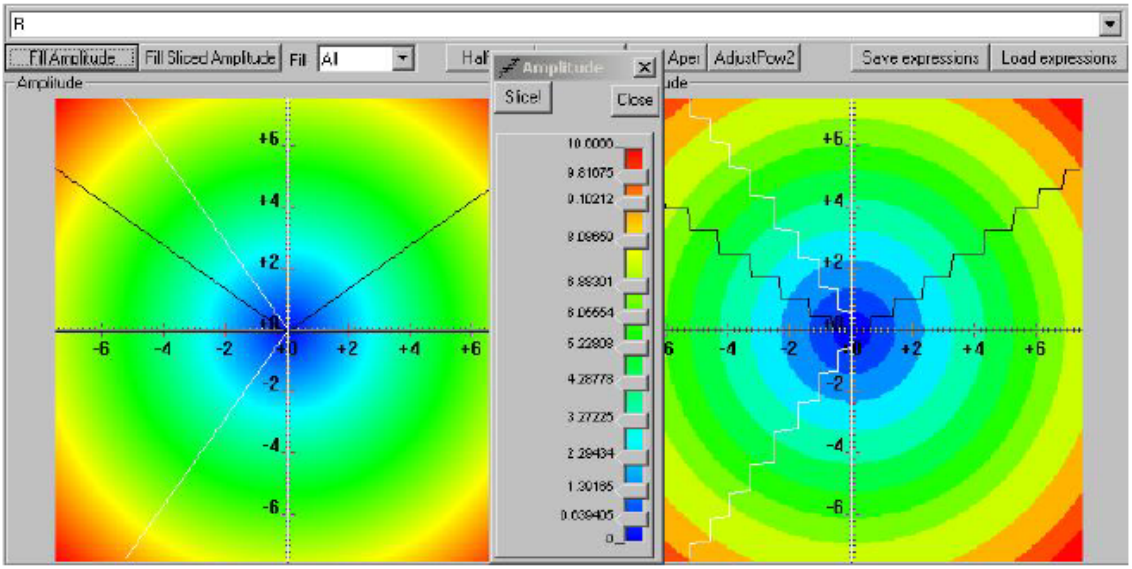
単位の変換には次の定数を使うことができる。
mm: 1
um または **μ m**: 10^{-3}
nm: 10^{-6}
cm: 10
degree: $\pi/180$
radian: $180/\pi$

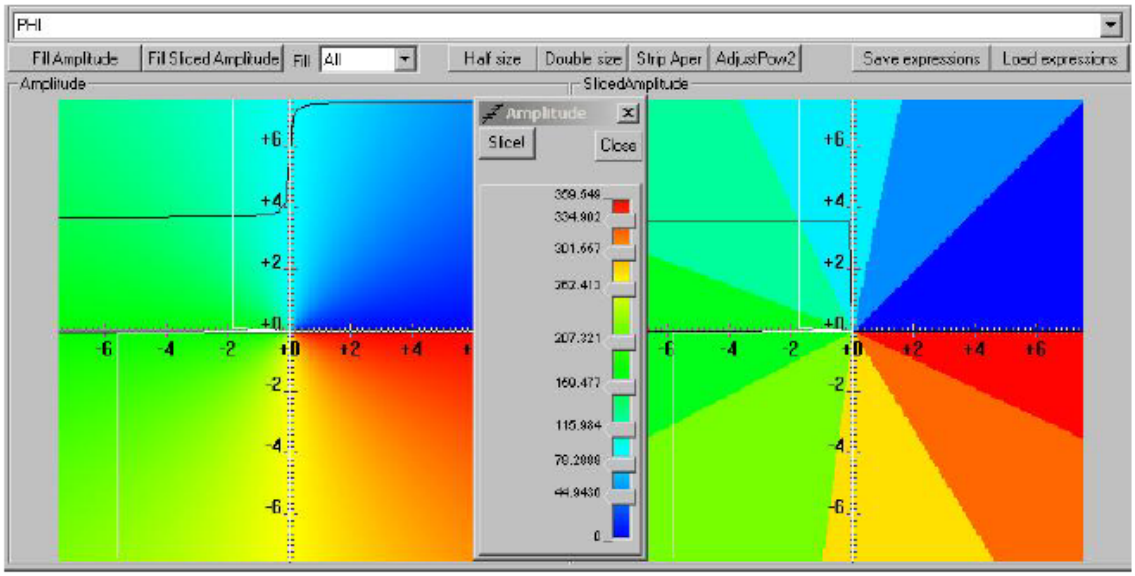
mm, **um**(または **μ m**), **nm**, および **cm** の定数は、長さの単位変換を容易にするために、プログラムに用意されている。*DiffractGUI* はすべての長さを mm で内部に収容し、したがって変換には、たとえば 123um を mm に変換する場合には、単に 123*um とタイプすればよい。12.34mm を um に変換するには、12.34/um とタイプすればよい。同様に、degree と radian 定数は degree から radien に、またその逆に変換するのに便利な方法を提供する。10°を radien に変換するには、10*degree とタイプすればよい：1.23radian を度に変換するには 1.23*radien とタイプすればよい。これらの定数を使うことは与えられた表現を変換する時間を短縮する。

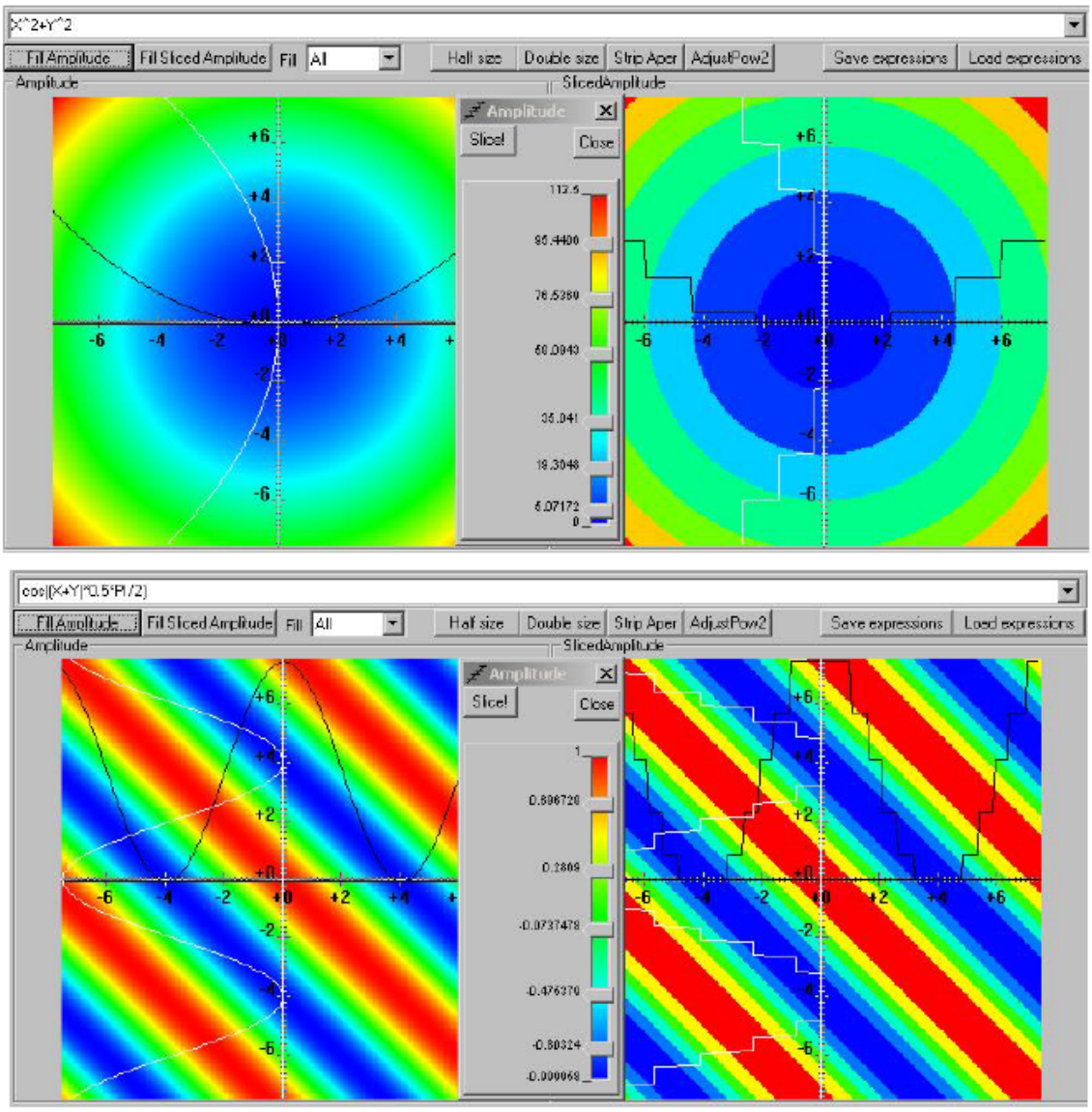
他の常数または変数を数式表現に定義してはいけない。そのような事をすると、”unknown parameter”というエラーメッセージの原因となる。

数式表現の解釈が終われば、その数式は combo-box のリストに加えられ、今後訂正・計算が行われる。しかし、Mask_Editor が閉じられると、数式リストは廃棄されてしまう。この数式リストは **Save expression** ボタンでディスクに保存し、次のシミュレーション演算では **Load expression** ボタンで再読み込みしなければならない。

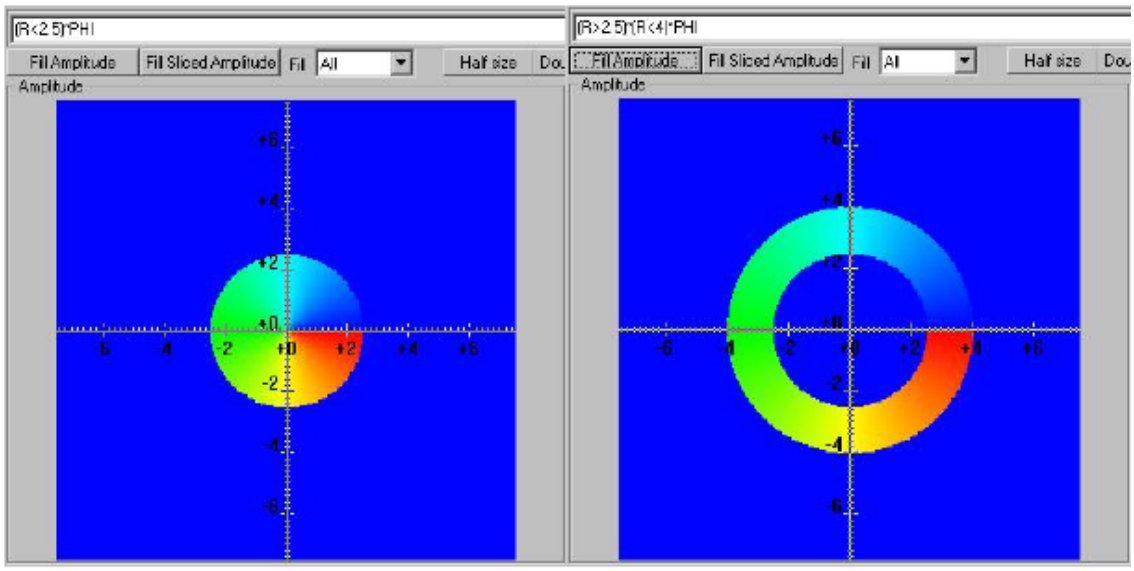
次の例は一般的に興味深いいくつかの場合について、Mask_Editor の **Fill** 関数の使用を示している。マトリックスを数字または数字のある範囲で満たすためには、単に数字（または上述の定義された変数）を expression box にタイプし、下図に示すように、**Fill** ボタンの一つをクリックすればよい。（ボタンの代わりに Pop-up menu 入力の **Fill!**を使ってもよい)。

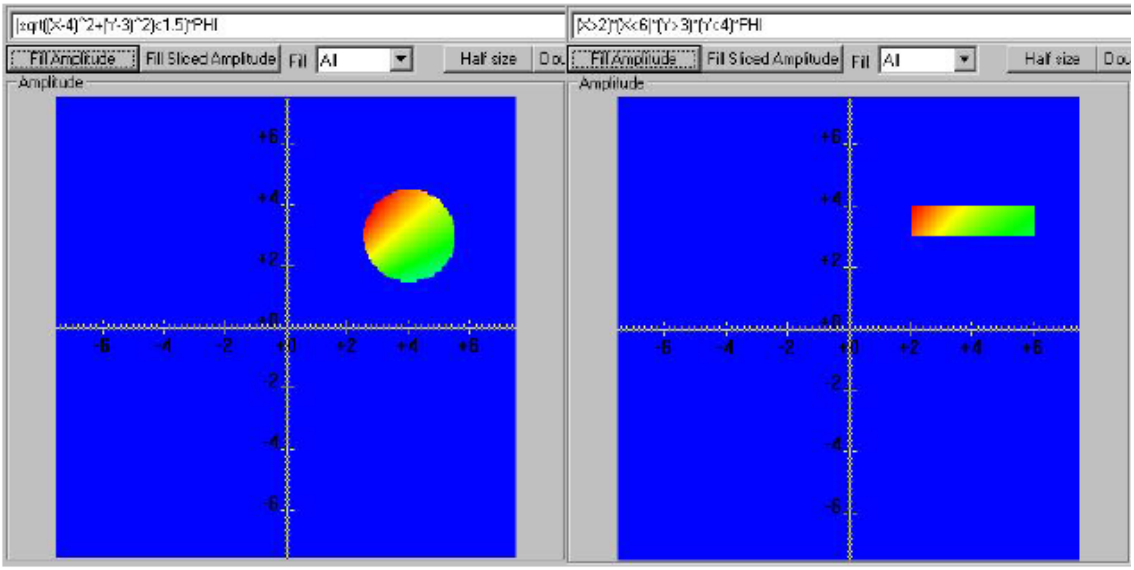

R
Fill Amplitude
Fill Sliced Amplitude
Fill
All
Aper
AdjustPow2
Save expressions
Load expressions
Amplitude
Slice!
Close
10.0000
9.81075
9.10212
8.09650
6.89301
6.06554
5.22809
4.26778
3.27225
2.29434
1.30185
0.039405
0


PHI
Fill Amplitude
Fill Sliced Amplitude
Fill
All
Half size
Double size
Strip Aper
AdjustPow2
Save expressions
Load expressions
Amplitude
SlicedAmplitude
Slice!
Close
359.648
334.902
301.667
252.413
207.321
160.477
115.984
70.2888
44.9430
0

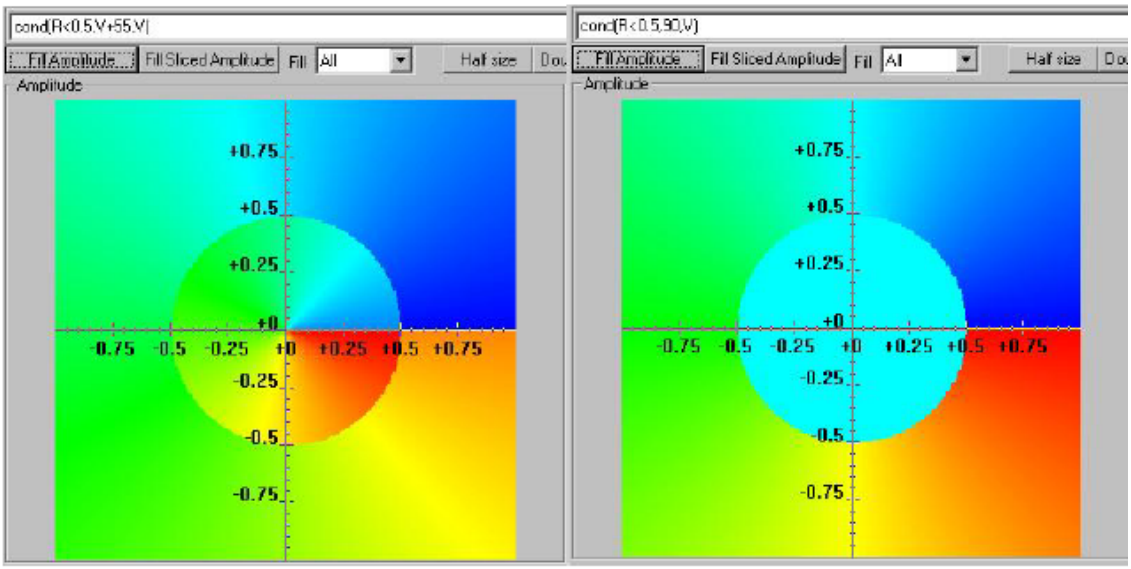


上の例では、マトリックス全体はそれぞれの成分に入力されている。しかし、比較演算子(>, <, =)を用いて簡単な形を作ることができる。次の例でこの方法を説明している。これらの例で、形の外側の領域は 0 が入力されている。同じような方法は、既存のマトリックスの一部を使う場合にも適用することができる。

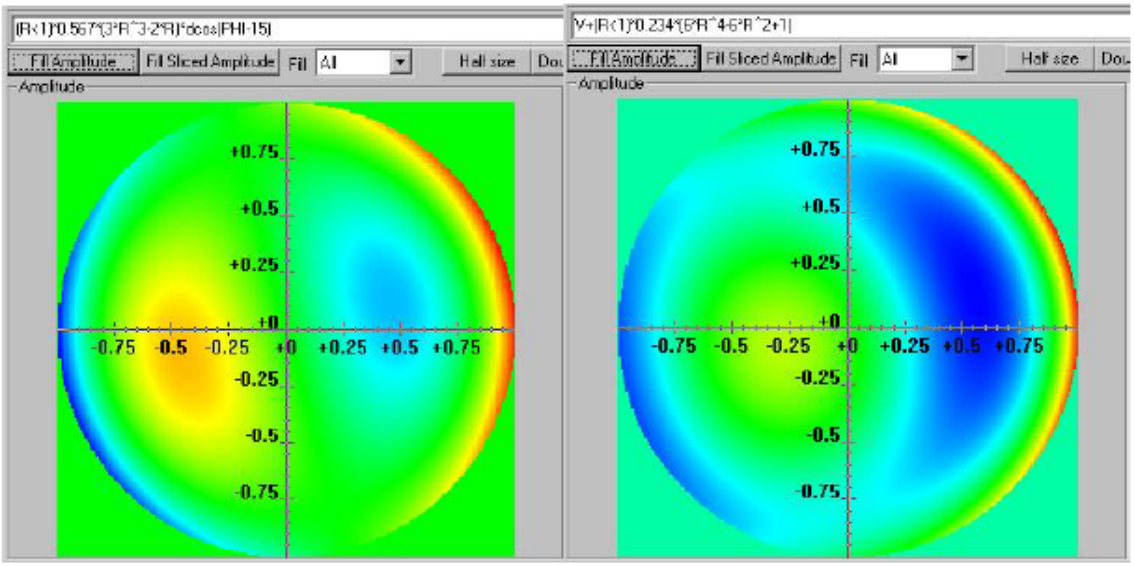




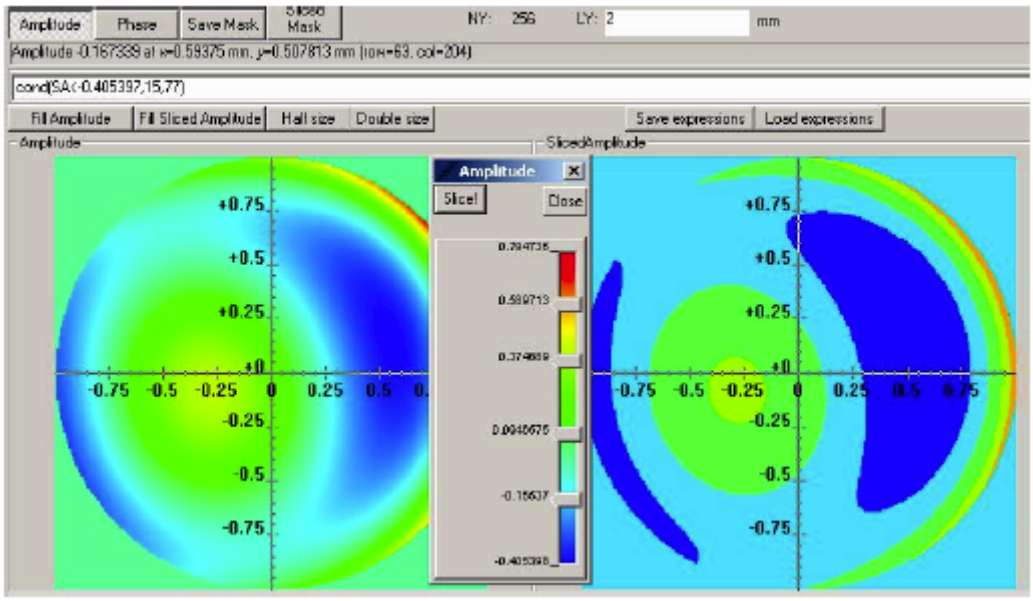
既存のマトリックスのある 1 部分（または全体）を修正しようとするとき、同じような状況が起こる。これには予め定義された変数**V** と **cond** 関数を使う。マトリックスが初めに **PHI**（上図を見よ）が入力されている場合を考え、ユーザーは中心の周り半径 0.5mm の領域を変更し、他の部分はそのままにしようとしている。次の例はどのようにしてこの最終結果を有効にするかを示している。下の左側の例では、中心に 55 の値を加え、右の例では中心領域全体に 90 の値を入れた。どちらの場合も、中心の円の外側は変えていない。



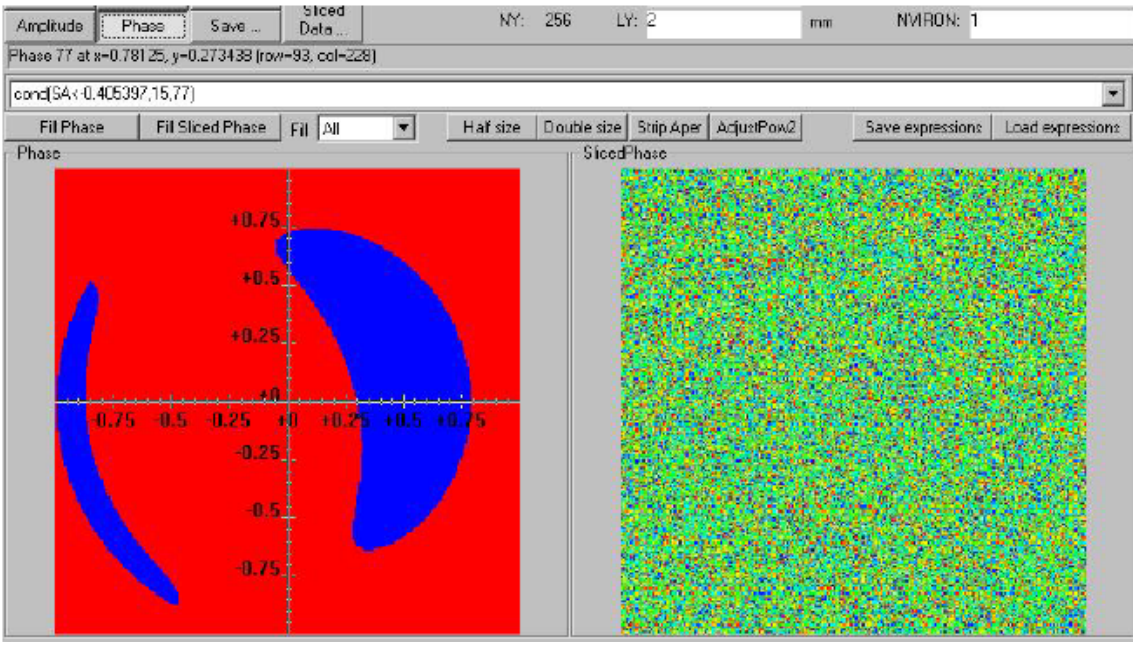
次の例では、Zernike 多項式（3,1）と（4,0）を基にした位相マスクを作る。マトリックスは最初、左側の図に示すように、(3,1)多項式で入力し、それから最終マスクを得るために(4,0)多項式を加えた。ここで、$C_{31}=0.567, \phi_{31}=15^{\circ}$および$C_{40}=0.234$である。マスクは半径$R=1$の円形領域に限られている。



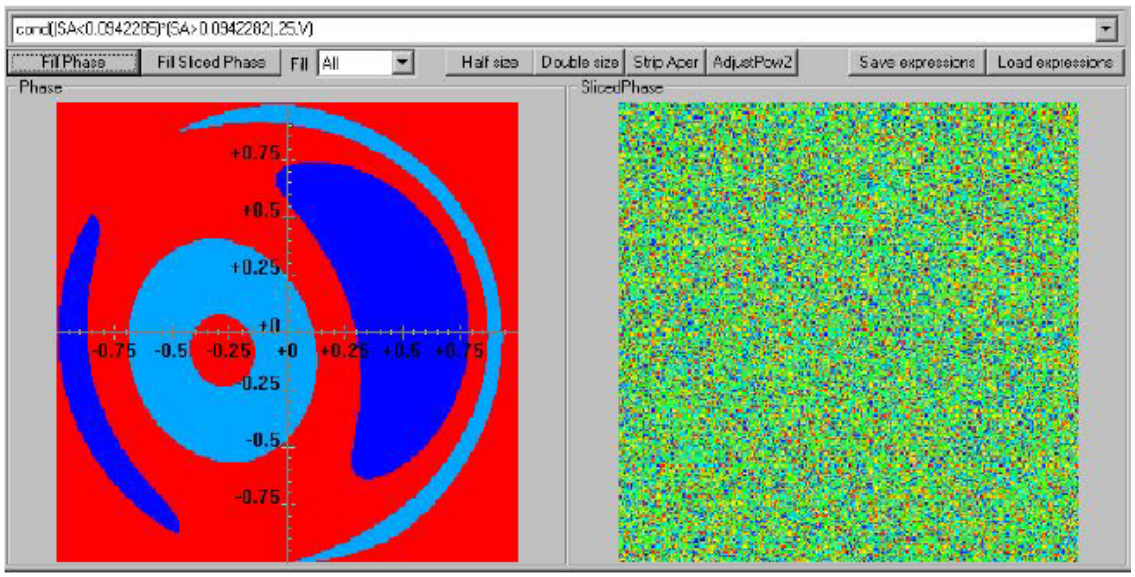
上記の振幅からスタートして、代表的な位相マスクを作ってみよう。最低振幅の領域（次の図で、スライスされた振幅の濃い青の領域）に15°の位相を持たせ、その外側に77°の位相を設定した。



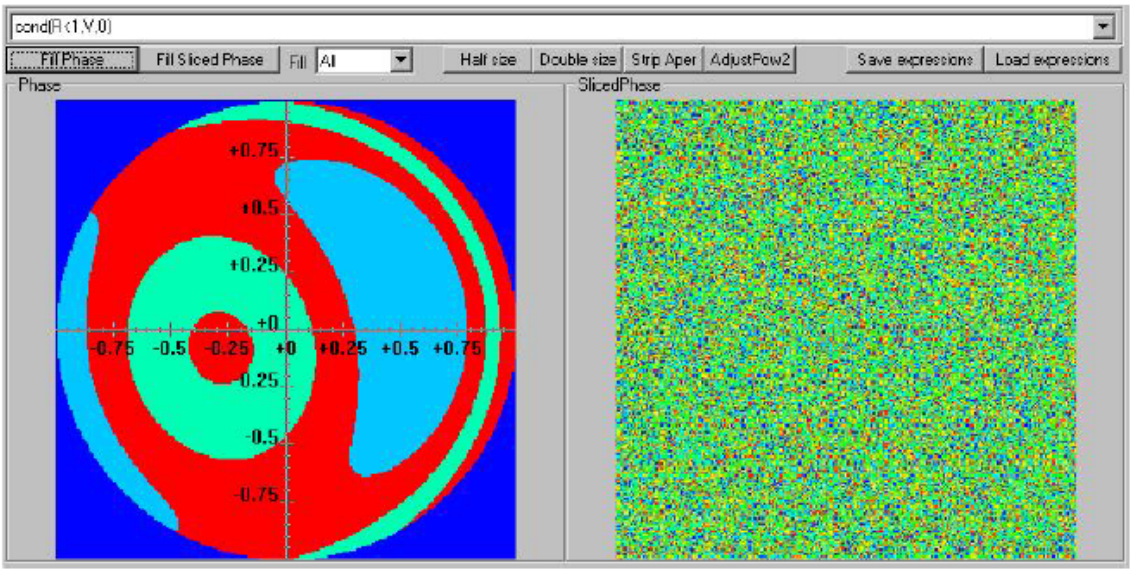
先ず、最低振幅の領域（dark blue の領域）の内部におけるスライスした振幅の値は－0.405398 であることに注目し、スクリーンに位相マトリックスを表示するため **Phase** ボタンをクリックする。この時点では位相マトリックスは任意の値を持っているが、下に示すように入力することができる。



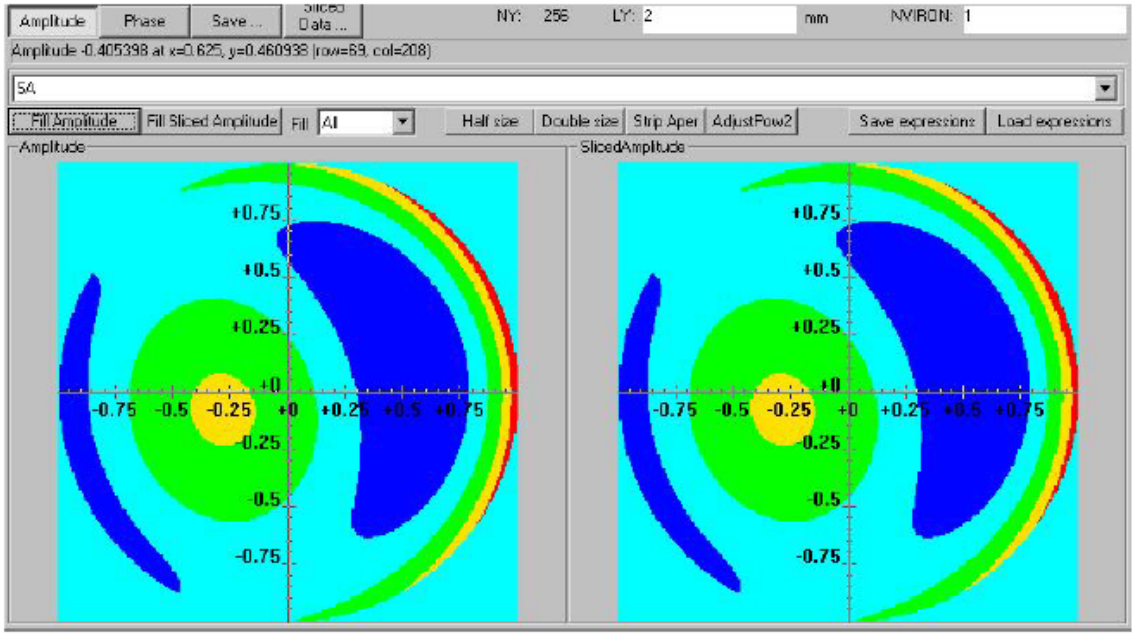
そこで、25°の位相をスライスされた振幅の緑色の領域に設定するが、この領域の外部の領域は変化せずにそのままとする。指定された領域の内部のスライスされた振幅は 0.0946575 であることに注意されたい。



0.0946575 が目的とした値であったため、比較 operator（＝）を使わなかったことに注目するのが大切である。実際の値は 0.09465745 と 0.0946576 の間にあるが、われわれは取り扱いを簡単に知るため、粗い限界の 0.0946574 と 0.946576 を用いた。同様に、位相マトリックスは他の振幅領域に入力される。最後に、$R<1$の領域は変えないで、半径Ｒ＝１の外側に位相を 0 と設定した。



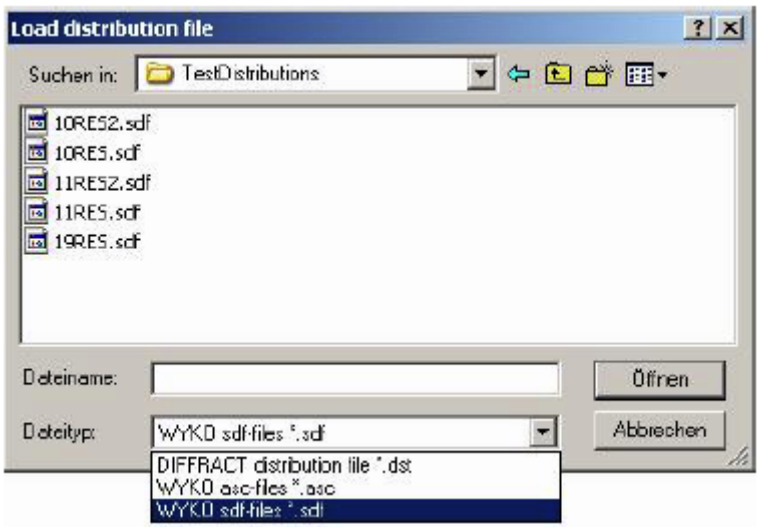
Save Mask ボタンをクリックしてマスクをディスクに保存する前に、(**Fill Amplitude** をクリックして）スライスした振幅を振幅マトリックスにコピーしなければならない。



別の方法として、(同じ方法で) 位相マトリックスをスライスした位相マトリックスにコピーすることができ、その後 **save Sliced Mask** ボタンをクリックする。

15.2 WYKO ファイルフォーマット(+.asc, *.sdf)で保存された分布の導入(Importing distributions stored in WYKO file format(.asc, *.sdf))

このような外部ファイルを読み込むとき、まず Mask Editor の Load Beam ボタンをクリックすると、次のような File Open dialog box が現れる。



この dialog box の中で、希望のファイル形式（上の例では*.sdf ファイル）を選び、ファイルリストから希望のファイルを選ぶ。次に、Enter キーを押して box を閉じる。この後、*.asc または*.sdf の場合には、Mask Editor はデータを外部ファイルからプログラム内部の位相マトリックスに読み込み、その後のデータ操作（たとえば Fill-operation, slicing など）が使用可能となる。振幅の値は一般的に 1 と設定される。もちろん、もしファイル形式が DIFFRACT の分布ファイル*.dst であれば、振幅と位相の両方がファイルから抽出される。次に、どのような場合でも、振幅、位相マトリクスは対応するスライスマトリックスにコピーされる。

WYKO 形式のファイルはヘッダーを持ち、その後にデータアレイがあることが望ましい。次の節ではそのようなファイルを読み込んだとき、関連するパラメーターの詳細を含めて、*DiffractGUI* に望ましい形式について述べる。

15.2 WYKO *.asc ファイルの望ましい形式（Expected format of the WYKO.asc files）

確認のために、*DiffractGUI* はテキストの第 1 行の”WYKO ASCII”の文字をチェックする。もし、これが第 1 行の最初に置かれていないときには、そのファイルは拒否される。次の行は先ず token に分割される。token は space で区切られた連続した文字であって、space は区切り文字として使われている。もし、3 個より少ない token しか 1 行にないときは、その行は無視され、インタープリーターは次の行に jump する。実際に、ヘッダーのすべての行は”X size”と”Y size”があるかどうかをチェックされる。もしこれらが見つかると、”X size”または Y size”の次にある数値を読み込み、それぞれ NMAX または NMAY に割り当てる。もし行の最初の token が”OPD”だとすると、ヘッダーの終わりと考え、次の NMAX 行がデータ値を表していると考える。これらの行のそれぞれは、TAB で区切った NMAY 個のデータ値を含んでいる。存在しない値は 0 で置き換えられ、それ以上に付け加わった値は無視される。“OPD”ブロックを読み込んだ後、その後にどのような行やアレイが読み込みファイルに存在しても、インタープリーターはストップする。

WYKO *.sdf ファイルの望ましい形式（Expected format of the WYKO *.sdf files）

sdf-ファイルは ASCII またはバイナリーファイルとして読み込まれる。*DiffractGUI* はファイルの最初の 4 バイトを考慮して、自動的にタイプを決定する。もしこれらが”aBCR”なら ASCII 形式であるとし、そうでなければバイナリー形式と考える。

Bainary ＊.sdf files

望ましいヘッダーは次の構造を持っていなければならない。

8 文字 －最初の 4 文字は”aBCR”でなければならない。
10 文字－ManufacID (無視)
12 バイト－Create Date（無視）
12 バイト－ModDate（無視）
2 バイト－NumPoints, この値は NMAX に設定される。
2 バイト－NumProfiles、この値は NMAY に設定される。
8 バイト－XScale（無視）
8 バイト－YScale（無視）
8 バイト－Zscale（無視）
8 バイト－Zresolution（無視）
1 バイト－Compression（無視）
1 バイト－Data Type（無視）
1 バイト－Check Type（無視）

ヘッダーの直後、データがあるものと思われる（データ値当り 1 バイト)。したがって、*DiffractGUI* は NMAX×NMAY バイトのデータをヘッダーの後で読み込む。最初の NMAX バイトは最初の行、次の NMAX バイトは第２行、・・・のように並んでいると考えられる。

*ASCII *.std files*

この種のファイルは行ごとにチェックされ、第 2 行からスタートする。最初の行は最初の 4 バイト”aBCR”を持たなければならない。そうでないと *DiffractGUI* はバイナリー形式の *.sdf ファイルと考える。
これらのファイルのヘッダーは次のようになっているはずである。

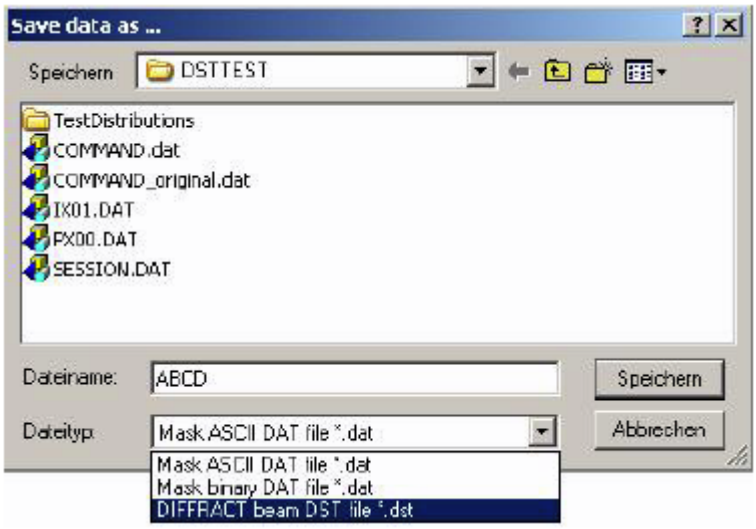
aBCR????
ManufacID = 文字
CreateData = 数値
ModDate = 数値
NumPoints = 数値
Xscale = 数値
Yscale = 数値
Zscale = 数値
Zresolution = 数値
Compression = 数値
DataType = 数値
CheckType = 数値

*

解析の途中では、それぞれの行はスペースで区切られた token に分割される。もし最初の token が”NumPoints”または”NumProfiles”なら、数値 token はそれぞれ NMAX あるいは NMAY にセットされる。もし、最初の token がアステリスク（”*”）の場合、ヘッダーはこの行で終わりと考えられ、*DiffractGUI* はデータ行として次の NMAX×NMAY 行を読み込もうとする。これらのデータ行のそれぞれは 1 つの数値 token を持ち、これは倍精度浮動小数点数値（double precision floating point value）として *DiffractGUI* が読み込む。(ヘッダーの最初の token に対する他の名前はインタープリーターによって無視される。)

15.3 データを DIFFRACT 分布ファイルとして保存（Storing data in DIFFRACT's distribution file format(*.dst)

これは Mask Editor の **Save...**または **Save Sliced Data** ボタンによって行われる。次のような File Save dialog box が開かれる。

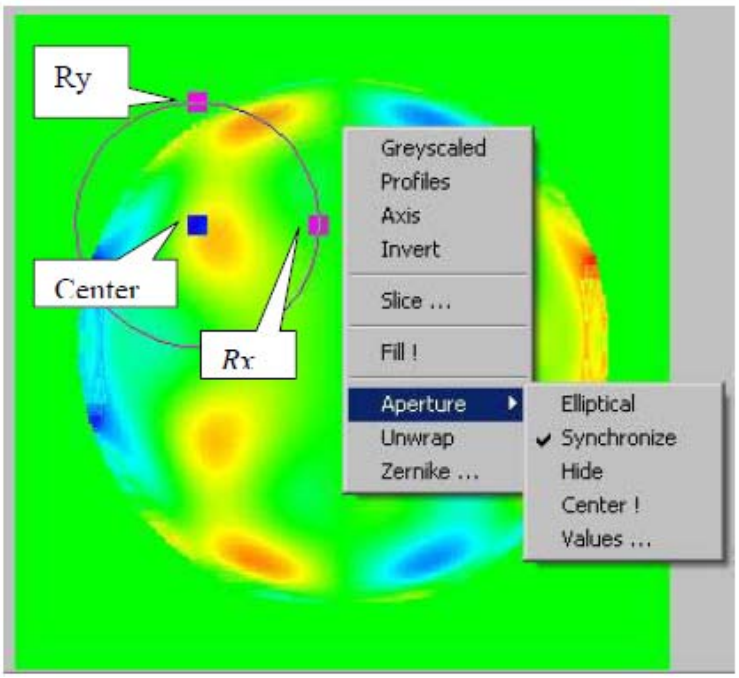


上図に示したように、希望のファイルを選ぶか、新しいファイル名(この例では ABCD)をタイプするか、あるいは上書きして保存する既存のファイルを選び、**Save** をクリックするか、*Enter* キーを押す。

15.4 ユーザーが定義した開口（User defined apertures）

下図に示すように、Mask Editor はデータマトリックスに円形または楕円形の開口を設定することができる。この開口は Fill、slice、あるいは単にマトリクスのサイズ（すなわち

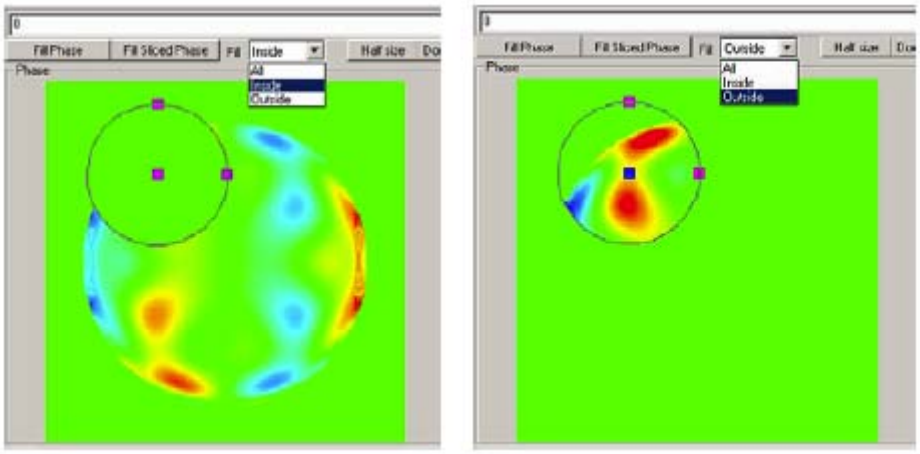
NMAX と NMAY）を任意の値に決めるとき、境界を示す領域として使うことができる。



開口は 3 個のパラメーター：*Rx, Ry* および *Center* で定義され、これらはマウスカーソルでのみ変更できる。それらの一つを変えるときは、マウスカーソルを対応する正方形上に置き、（上図を見よ）、左マウスボタンを押し下げたままマウスを動かせばよい。操作を終えるにはマウスボタンを離せばよい。右ボタンで開く Pop-up menu で plot をクリックすると、次のような option を含む submenu **Aperture** が開く。

- **Elliptical** － 開口を楕円形にする。基本的には、*Rx* と *Ry* を独立にセットできる。
- **Synchronize** －この flag をセットすると、スライスされた map の開口は自動的に non-sliced map の開口の変化に追従する。
- **Hide** － スクリーンにおける円形（または楕円形）開口の描画から遮断する。
- **Center !** － この command が起動されたとき、開口の中心が map の中心の(0,0)にセットされる。
- **Values** － 次のような dialog box を開き、開口パラメーターを明示し読むことができる。

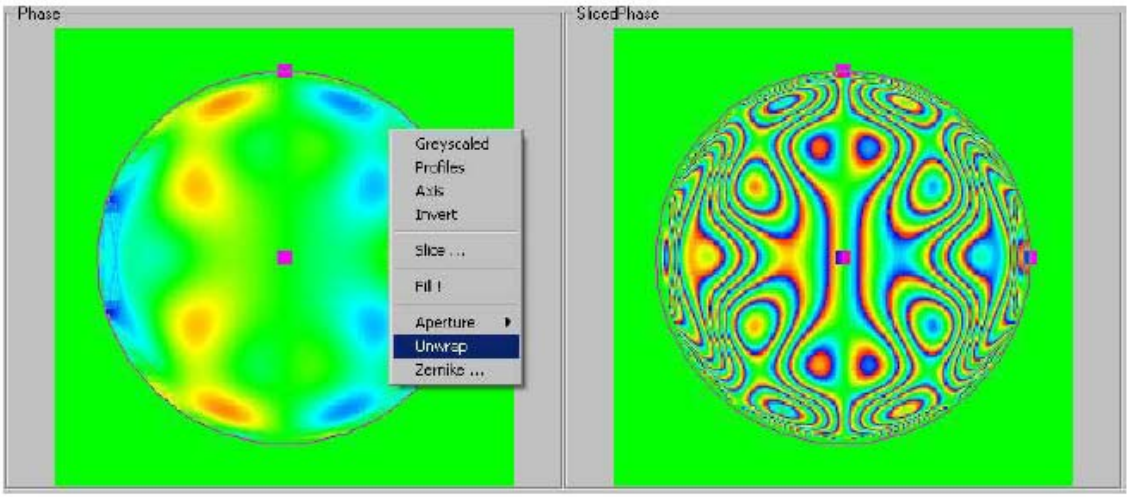
開口の一つの応用は、Fill-機能が動作する面積を定義することである。３つの option が可能である。すなわち、**All**（開口を無視する）、**Inside**（開口の内部の点だけを考える）、および **Outside**（開口の円または楕円の外にある点だけに Fill operation が有効）である。次の図は **Inside** および **Outside** の fill モードを示している。さらによく理解するためには、次の図を前の図、すなわち fill する前の初期 map と比較されたい。.



同じ機能は slice operation でも有効である。通常、もし synchronize flag がセットされている（これは default 条件である）と、slice した開口と non-slice の開口は同一である。しかしもし違った条件の場合は、slice するときは non-sliced map の開口のみを考えることになる。

15.5 2π の位相不連続を解消するための Phase-unwrapping（Phase-unwrapping to remove 2π phase discontinuities）

三角関数の周期性のため（また光の振幅分布は通常複素数として保存されるため）、位相の値は－180°と+180°の間に限られる。したがって、中程度または大きい収差のある波面の場合、下図右のように不連続線（位相が－180°から+180°へ飛ぶ線）が現れる。このような位相の不連続線は、pop-up menu の命令 **Unwrap** を用いれば、下図左のように解消される。（位相 map のみが unwrap され、振幅 map の場合には menu option はグレイになって実行できない。）もし分解能が粗過ぎる場合（たとえば NMAX, NMAY が小さい場合）、換言すれば、もし位相の振動が 360°より大きい値について速く振動しすぎる場合は、この unwrap 命令は実行できない。

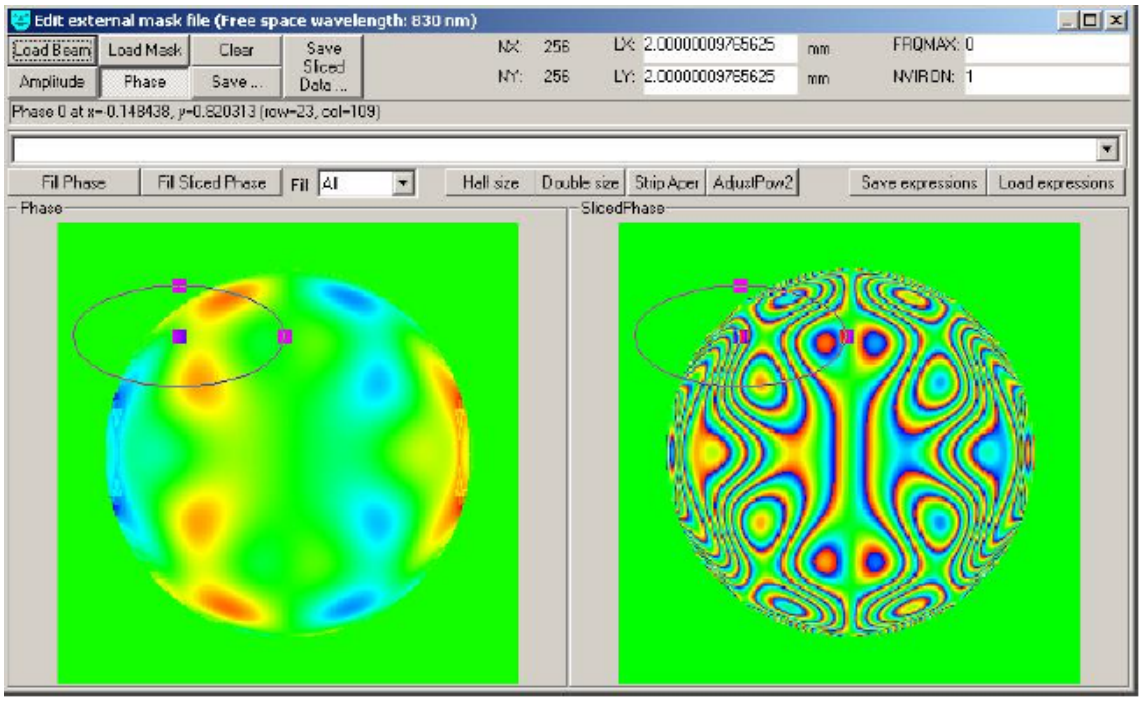


15.6　データアレイのサイズ変更（Resizing data arrays）

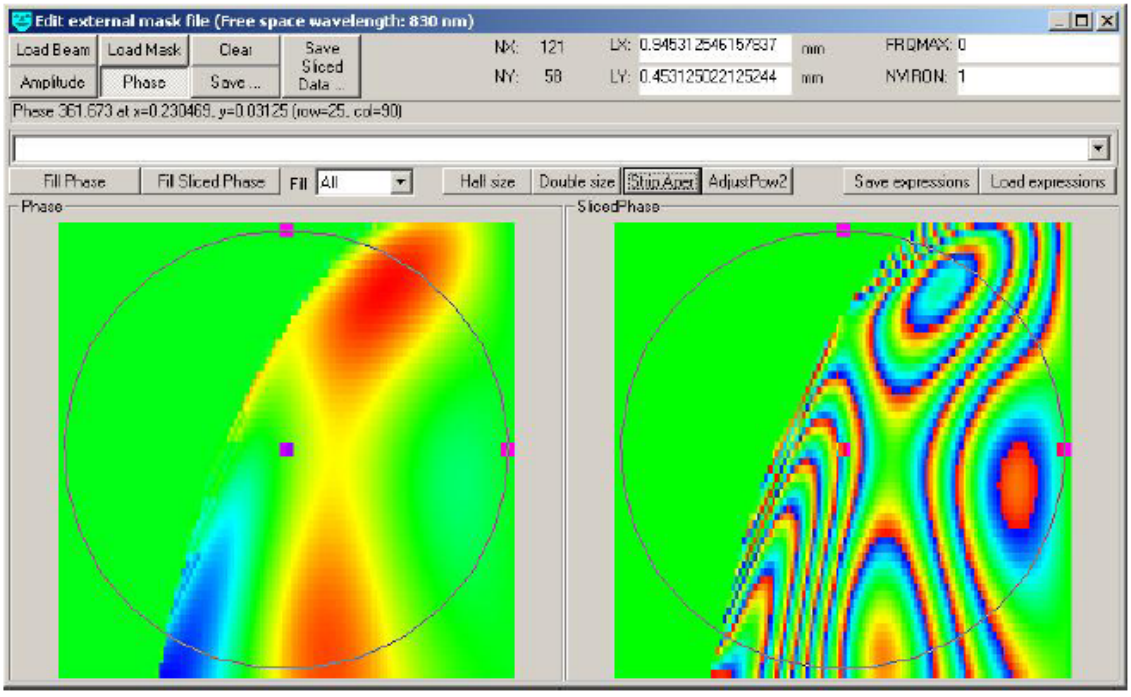
すべての読み込まれたファイルが正方データマトリックスとなるわけではない。それは NMAX はいつも必ず NMAY と等しいわけではないということである。さらに、読み込まれたファイルの小さい面積だけが DIFFRACT による今後の計算に使われることもある。

DIFFRACT は NMAX が NMAY と等しくない場合も取り扱うことができるが、読み込まれたデータを DIFFRACT 分布ファイル(*.dst)、あるいは XOBJ ファイル(*.dat)に記録する前に、それらは両方とも 2 の冪数であることが望ましい。DiffractGUI Mask Editor は 2 個のボタン **StripAper**,および **AdjustPow2** によって、マトリックスサイズ（すなわち NMAX と NMAY の値）を変えることができる。（警告：これらの operation はマウスのクリックによって直ちに実行できるが、やり直しはできない。したがって、よく考えて使う必要がある。）

StripAper
すべてのデータマトリックスは、円形（または楕円形）開口を制限する正方形で定義された sub-matrix で、non-sliced map（左側の map）を置き換えることができる。NMAX と NMAY は新しい値にセットされ、この正方形の外側にあるデータポイントは削除され、もはや使えない。

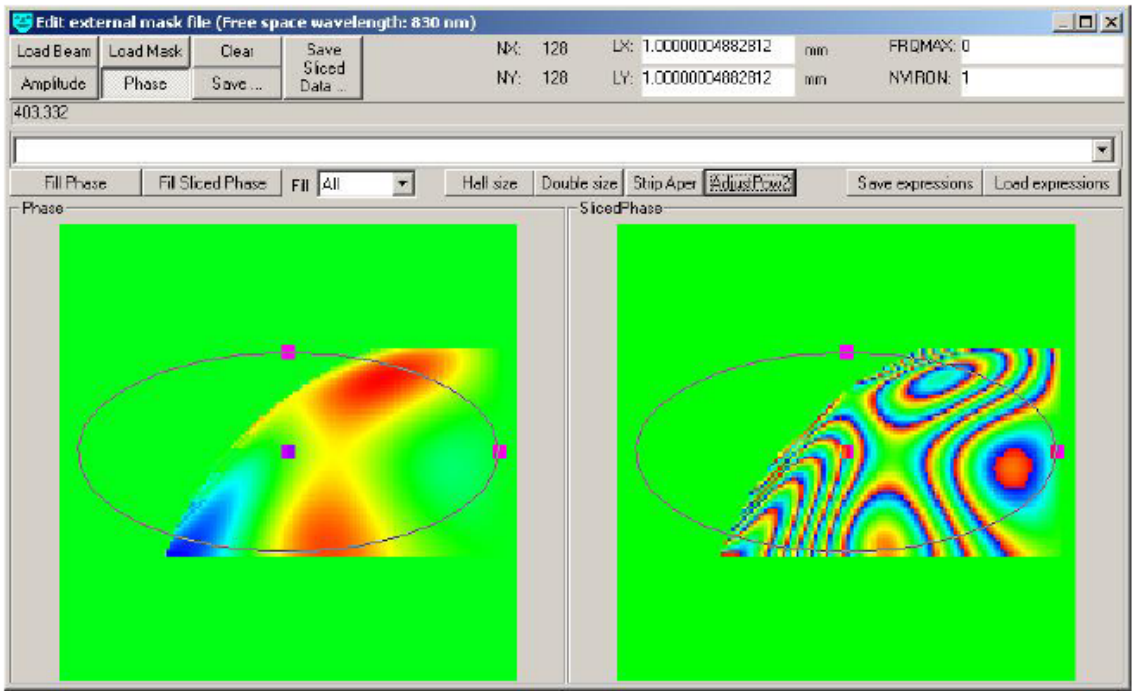


たとえば、上の図で示した map からスタートしたとき、**StripAper** 命令による結果を次の図に示す。



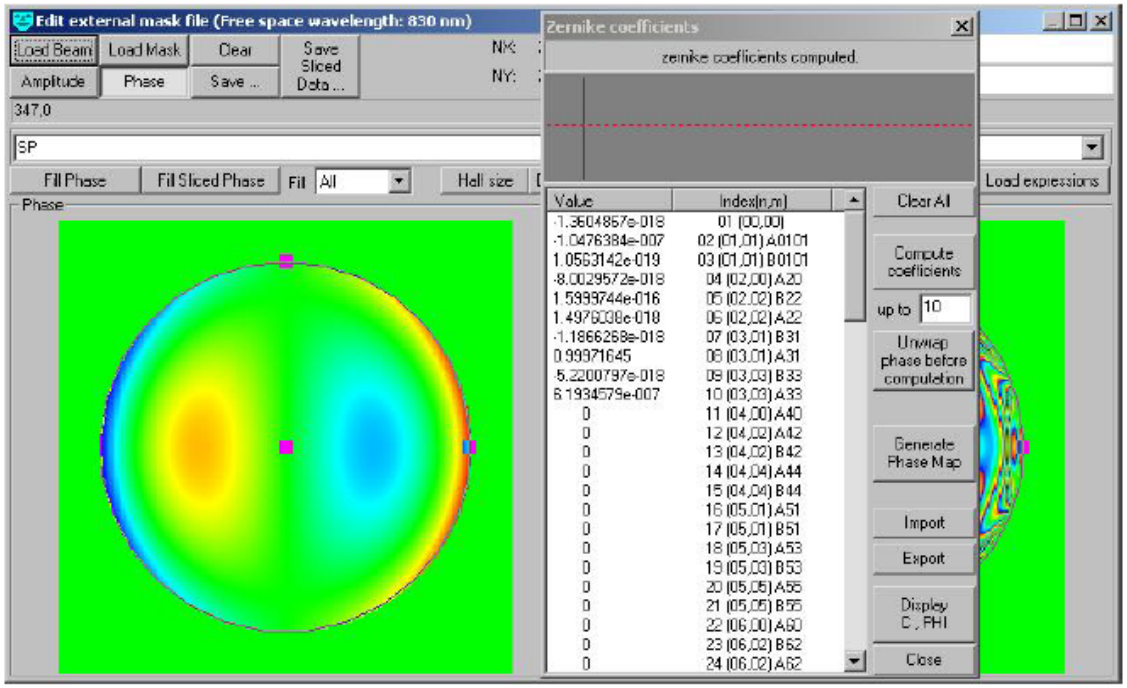
AdjustPow2

この command はデータマトリックスのサイズを次の 2 の冪数まで拡大する。このときデータの存在する領域の周りの新しい領域には 0 を入れる。たとえば、上図の位相マップは、AdjustPow2 command を起動すると次図のようになる。



15.7　位相マップから Zernike 係数の抽出（Extracting Zernike coefficients from phase distribution）

pop-up menu の **Zernike...**command をクリックすると、下図に示すように、モード無しの dialog box が開く。これによって、Zernike 多項式を使って位相アレイを変更したり解析したりすることができる。



1 から 91 まで番号を付けられた最初の 91 個の多項式は、次の定義を使って求めることができる。

$$Z_{even\_j} = \sqrt{2(n+1)} \cdot R_n^m(r) \cdot \cos(m \cdot \theta) \qquad (m \neq 0)$$
$$Z_{odd\_j} = \sqrt{2(n+1)} \cdot R_n^m(r) \cdot \sin(m \cdot \theta) \qquad (m \neq 0)$$
$$Z_j = \sqrt{(n+1)} \cdot R_n^m(r) \qquad (m = 0)$$

ここに radial polynomial（極座標を使った多項式）は次のように定義されている。

$$R_n^m(r) = \sum_{s=0}^{(n-m)/2} \frac{(-1)^s \cdot (n-s)!}{s![(n+m)/2-s]![(n-m)/2-s]!} \cdot r^{n-2s}$$

指数 j は Zernike 多項式を特定し、それぞれの j には一対の数字(n,m)が付与され、これが対応する radial polynomial を定義している。この dialog box は、リストボックスの右側の欄に、この係数をさらに親しみやすい記号 **A_{nm}** または **B_{nm}** と共に表示する。A_{nm}, B_{nm} の代わりに、DIFFRACT はさらに実際的な **C_{nm}**, **PHI_{nm}** option を使用する。dialog box はこれら 2 つの表現を **Display C, PHI** ボタンで切り替えることができる。

dialog list box の左の欄に 91 個の Zernike 関数のそれぞれに対応する係数 **a_j** を表示する。

数値はリストボックスで希望の entry を 2 回クリック（ダブルクリックではない！）することによって、キーボードで編集することができる。

Clear All ボタンにより、すべての $\mathbf{a_j}$ 係数を 0 とすることができる。

Export または **Import** ボタンにより、このマニュアルの Sec.13 で述べた形式の一つを用いて、aj 係数を外部ファイルへ読み込み、または読み出しすることができる（*Import/Export of Zernike Coefficients*）。

[**註**：MaskEditor の Zernike 対話ボックスで、Z1 関数（piston, a simple phase offset）が使われているが、DIFFRACT はこのこの係数を使わないことも注意すべきである。Import/Export 機能の主な目的はこれらの係数を DIFFRACT に渡すことであるため、Z1 係数は importing/exporting に際して無視される。もし必要があれば、Z1 はキーボードを使って手動で入力しなければならない。]

Generate phase map ボタンは、前もって定義された開口内の位相アレイを満たすのに用いられる。開口内部の各点は極座標 r_i, θ_i で表され、位相 Φ_i（度で表す）は次の式で計算できる。

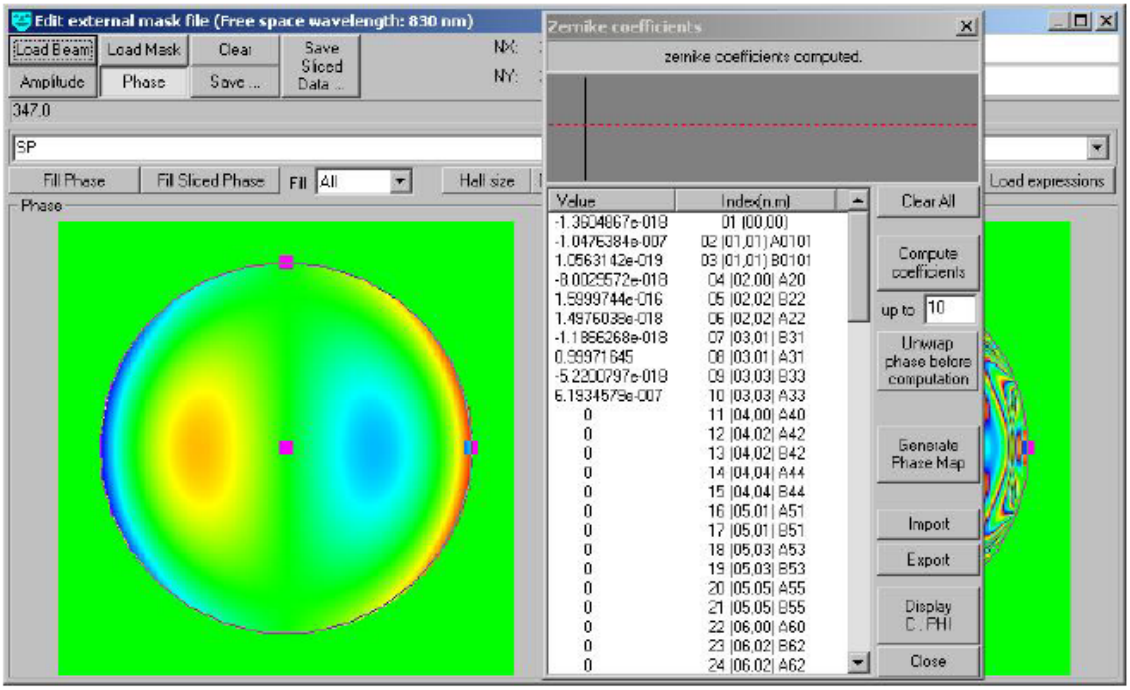
$$\Phi_i = \sum_{j=1}^{91} a_j Z_j(r_i \cdot \theta_i) \cdot 360$$

ここに $\mathbf{a_j}$ は dialog list box から得られる。開口の外では位相は 0 とセットされる。開口はメッシュの中心に中心を置いてはならないし、円形であってはならない。それぞれのピクセルの r_i 値は、開口の半径で基準化される。楕円開口の場合は、r_i 値は 2 つの半径 R_x, R_y のうち大きいほうで基準化される。1 例として、前のページの図は係数 a_8 が 1.0 にセットされた（$A_{31} = 1.0$）とき **GeneratePhase Map** ボタンで作られた位相マップであって、他のすべての係数は 0 にセットされている。

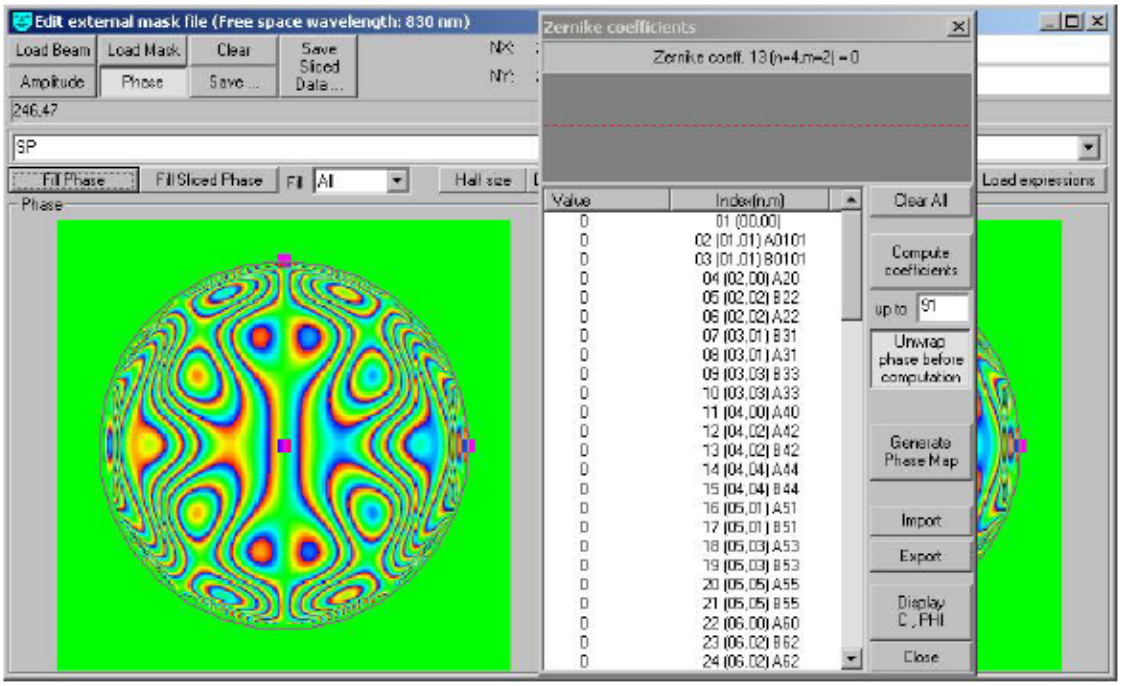
任意の位相 map があると、**Compute coefficients** ボタンは定義された開口の中にある既存の位相マップに最も近い波面を与える $\mathbf{a_j}$ 係数を求めるのに使うことができる。このアルゴリズムは開口全面（開口外部のデータは無視される）に対して最小二乗法でフィットさせることに基礎を置いていて、1 から **up to** edit box で規定される index までのすべての展開係数を計算している。（**註**：それぞれの抽出された係数について、アルゴリズムは開口内のすべてのサンプル点における一時的なデータアレイが必要になる。このメモリー消費のため、係数の最大次数を up to edit で制限したほうが良い。）一例として、前の例における位

相マップに係数抽出を適用した結果を次図に示す。

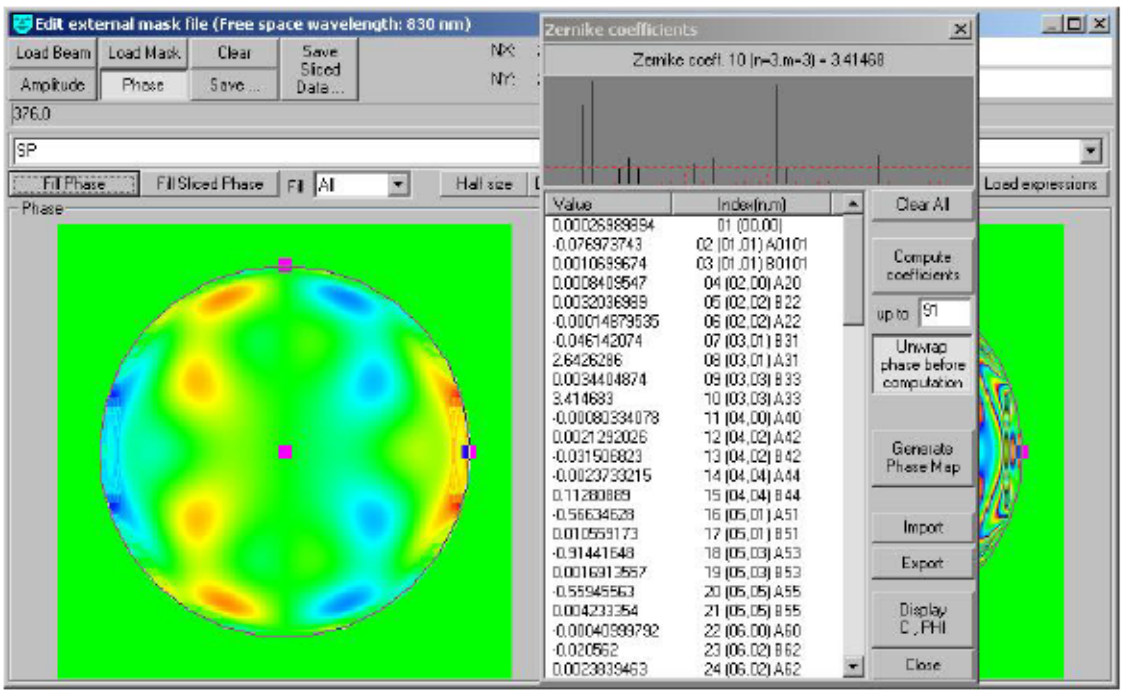
Zernike 係数のリストの上にある濃いグレイの矩形内に、Zernike 係数の絶対値が垂直の黒い線としてスペクトル状にグラフィカル表示されている。これは 2 つのタスクを実行している。一つはそれぞれの係数の貢献を他の係数と比較して容易に知ることができる。もう一つは、閾値（赤の点線で表示）を示し、これ以下の係数をゼロにして良いということを確認できる。この使い方を次の例で示す。



このスペクトル制限の一つの応用を示すために、次の図に示すようなさらに複雑な位相分布を考える。

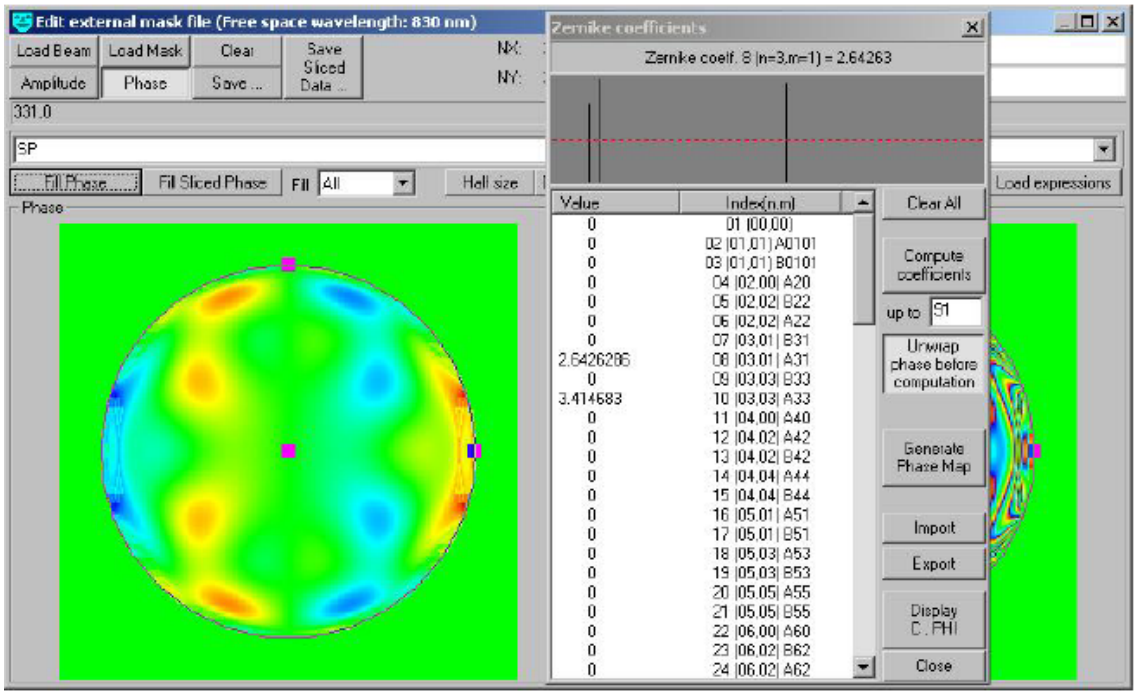


ここで、**Clear All** ボタンを使って、すべての係数をゼロとする。計算の前に位相分布は unwrap しなければならないので、**Unwrap phase before computation** ボタンを選択し、その後 **Compute coefficients** を使って計算を初期化する。その結果を次の図に示す。この例における係数のほとんどはかなり小さいが、それらのうち 3 つが明らかに目立っている。

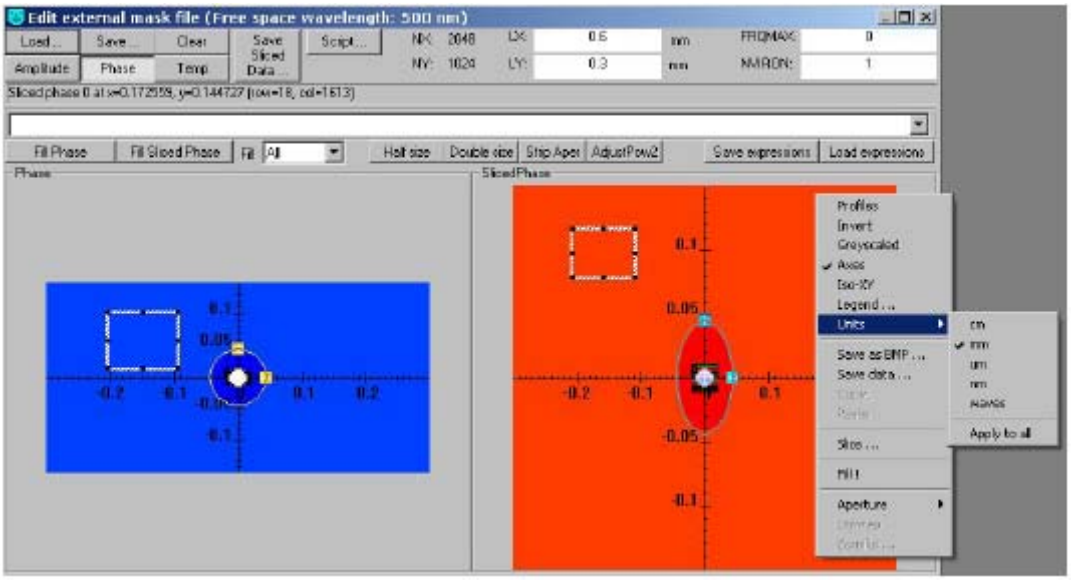


これらの係数のみを残し、他の係数をゼロに設定するために、マウスカーソルを使って赤い閾値ラインを正しく設定する。(左マウスボタンを押し、マウスカーソルを動かす。ボタンが押されている間、閾値ラインはカーソルに従う。) 絶対値がこの閾値以下のすべての係数は赤字で表示され、容易に認識できる。閾値を任意のレベルにおき、スペクトルの上に置いた左マウスボタンをダブルクリックすると、この閾値以下の係数はすべてゼロとなる。その結果を次の図に示す。

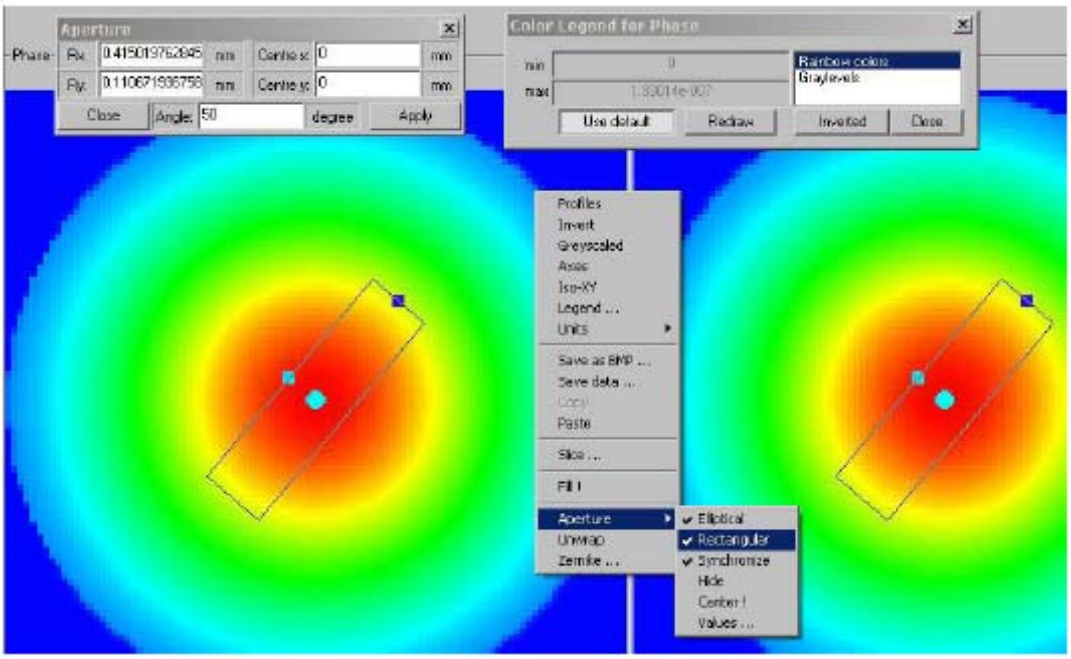
抽出のアルゴリズムの精度は開口の半径に強く依存し、また計算領域内のサンプル点の数に関係していることに注意されたい。最良の結果を得るためには、開口は円形で分布と正確に一致する半径を持つことである。さらに、Zernike 係数を抽出するためには最小二乗法が使われているので、抽出された係数は考えている分布を正確に表してはいないが、可能な最良の fit が得られている。もちろん、これは評価された結果の不正確さに加わる数値の丸め誤差とは別のものである。



BitmapView と同じように、MaskEditor の内部では、長さの単位と Iso-XY オプションはそれぞれの表示を個々に、下図に示すように選ぶことが出来る。



また、Tracker-Rectangles は、マスクの周りに動かしたり、クリップボードへ、またはクリップボードから、データをコピーしたり、貼り付けたりすることが出来る。次の図は、color regend を変更したり、矩形開口を作ったりすることが出来る。Tracker-rectangle と Iso-XY の詳細については BitmapView の項を参照されたい。



もし Rectangles オプションをチェックして、Elliptical オプションをチェックしないときは、開口は常に矩形となる。開口を回転させるには、Aperture dialog box の tilt angle に

タイプして（上図を見られたし）Enter を押せばよい。そうでなければ、開口の中心にあるドットにマウスカーソルを置き、マウスホイールを回転すればよい。

Color regend dialog box と Tracker-Rectangles は BitmnapView で述べたように働く。この詳細については BitmapView の項を参照されたい。MarkEitor の種々の map については BitmapView と同様にズームやパンが出来、これは細かい部分を修正する場合に役に立つ。

15.8 MaskEditor 内の算術演算（Arithmetic Parser Inside the MaskEditor）

上に述べた一般的な演算のほかに、MaskEditor の内部では次のような前に定義された付加変数と関数についての算術演算を行っている。

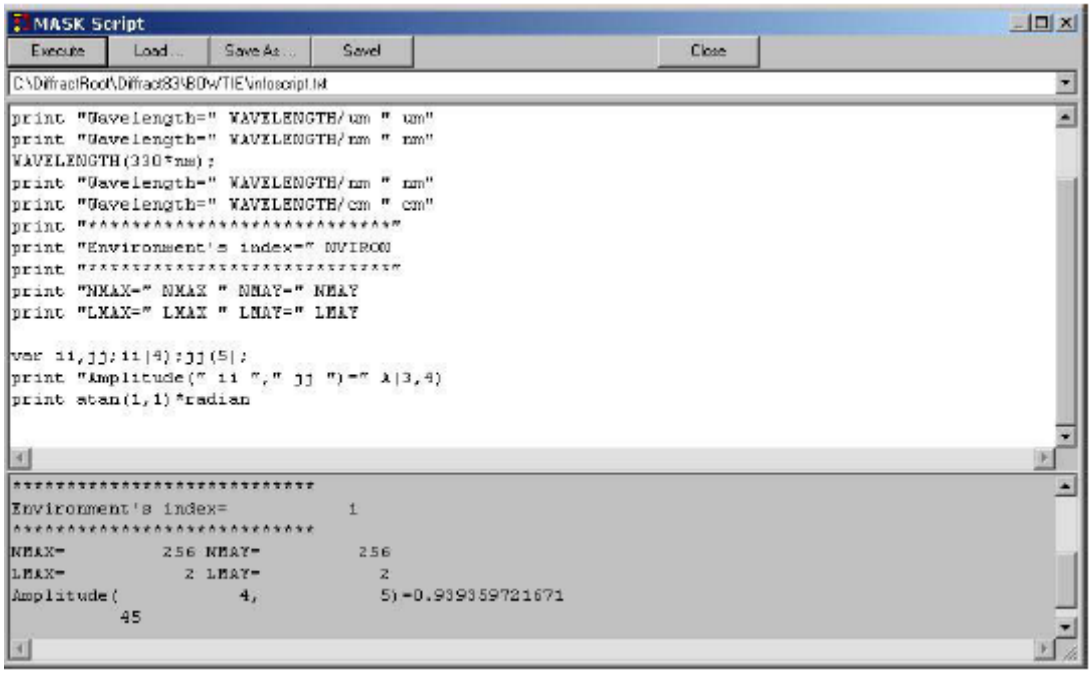
NMAX-NMAY-LMAX-LMAY-WAVELENGTH。

WAVELENGTH は常にｍｍで表示されていることを注意されたい。同様に、LMAX と LMAY も mm で表示されている。これらは容易に前に定義した常数 um, nm, cm で割り算して容易に変換できる。

関数 **A(n,m), P(n,m) SA(n,m), SP(n,m), T1(n,m), T2(n,m)** （すなわち Amplitude, Phase, SlicedAmplitude, SlicedPhase, Temp1 および Temp2）もそれぞれの曲面の n 行, m 列の値を返す。反対に、**A(n,m,x), P(n,m,x), SA(n,m,x), SP(n,m,x), Ti(n,m,x), T2(n,m,x)**もそれぞれの局面の x について n 行、m 列の値を与える。

15.9 MaskEditor の scripts (Scripts in the Mask-Editor)

データを編集するための適度の可能性として、MaskEditor には簡単な文字入力原語が取り入れられている。算術演算機能と関連して、この機能は描かれた図表の個々の値に直接介入することが出来る。簡単なループ、条件付構造、および変数を作る能力は、仕事を遂行するために便利な手段を提供している。これは、もしこの機能が無ければ困難であったことであろう。Script を使う直接的な方法は、MASK_script dialog box を開くことであって、これは MaskEditor の Script…ボタンを押せばよい。こうすれば、Scrit は外部ファイルを導入する間に使うことが出来る。



上の図で分かるように、MASK_Script dialog box は main text を入力する window を持っていて、ここに script をタイプする（あるいは外部ファイルから導入する）ことが出来る。また小さな出力 window と直前に使ったファイル（MRU）を表示する連携ボックスを持っている。**Load and Save As…**ボタンは script を導入または記録したファイル名をを表示し、また **Save!**…ボタンは MRU に表示したファイルを（確かめずに）上書きする。これは開発途中では便利であるが、そうでなければ注意して使わなければならない。

Execute ボタンは script を内部 interpreter を通して実行することを命令する。

基本的な文法は簡単である：それぞれの行は、コマンドキーワード（下に示す）の一つで始まり、その後に 1 つまたはそれ以上の要求されるパラメーターが続く。行の終わりには";"が置かれる、したがって";"で区切れば 1 行に多数のコマンドをおくことができる。キーワードとパラメーターは 1 次以上のブランク（空のスペース）で区切られろ。空の行、あるいは"//"で始まる行は無視される。したがって、"//"はコマンド行を挿入するのに用いられる。一般に、interpreter は大文字・小文字を区別できる。

次に示すキーワードは各行の最初におかれる。([　と]でつなぐのは任意である) :

var　　name 1 [, name 2, name 3,…]
0 またはさらに多くの変数で定義し、初期化する。　Script の中で普通の算術表現で用いるすべての変数は最初にこのコマンドで宣言する。　変数の名前は文字で始まり、alpha-

numeric 文字のみを含むものとする（たとえば、a-z, A-Z, および 0-9)。

ある値を変数として表すには variablename(value) を用いる。たとえば、 **myvar**, **hisvar** は **myvar** と **hisvay** と呼ばれる 2 つの変数を定義する。myvar の値をたとえば 1.23 に増加するためには、**myvar(myvar+1.23)**とする。　myvar の値をたとえば 1.5 だけ増加するときは、**myvar(myvar+1.5)**とする。この構造の他の重要な特徴は変数の現在の値に戻ることであって、これは下に示すようにループのプログラムを作りやすい。一般に、すべての変数は倍精度の floating number を持っている。それらを整数値に変換するには、**ceil( )** または **floor( )**を使う。

kill name 1 [, name 2, name 3,…]

この関数は変数 name 1, name 2, name 3…を script から取り除く。

stop
この命令は interpreter を停止させ、Script の実行を直ちに停止する。

print token, token

この命令は特定の token を（output window に）プリントする。token は算術的表現で表現は評価され、結果はプリントされる。あるいは、”と”で区切られたテキストで、このテキストはそのまま印刷される。MUST の token は互いにブランクで分離される。この理由で、算術的表現の token は、ブランクを含むことは出来ない。

例

print “hollo World”は、Hollo World をプリントする。
print “PI =”PI は　PI　＝　3.14159265359 をプリントする。
print A(3.8) は 3 行 8 列の振幅マップをプリントする。

```
if expression
...other command lines ...
else
...other command lines...
endif
```

or

```
if expression
...other command lines ...
endif
```

算術的表現は計算され、if-branch は結果がゼロでなければ飛び越される。もし、結果がゼ

ロの場合には、もしあれば else-branch が計算される。

例

var jj; jj(3); **if** jj=3; **print** "jj is three!"; **else**; **print** "jj is not three,"; **endif**;

" ; " はコマンド間の分離に使われる。同じコードは " ; " を使わないで次のように書くことも出来る。

```
var jj jj(3)
    if jj=3
    print "jj is three!"
    else
    print "jj is not three"
    endif
```

```
begin
...other command lines ...
  until  expression
```

```
while  expression
...other command lines ...
 endwhile
```

begin-until コマンドは begin と until のコマンドを exprssion の結果がゼロになるまで繰り返し継続することを意味している。while-endwhile は while と endwhile のコマンドを expression の結果がゼロで無い数になるまで繰り返し継続することを意味する。ループは無限のレベルの内部にネストされるが、それらは散らばることは無い。

For_loops は直接には装備されていない。しかし、これらは上記の **begin_until** あるいは **while_endwhile** の一方で校正することができる。

5 から始まって 10 で終わるループの例は次のようである：

```
var loopvar; loopvar(4);
while loopvar(loopvar+1)<11
print "loopvar is " loopvar
endwhile
```

```
var loopvar; loopvar(5);
begin
print "loopvar is " loopvar
until loopvar(loopvar+1)>10
```

特別なコマンドは、plot-map assignment である。その一般的な形は

plotmap identifier = expression

plotmap identifier として、**A, P, SA, SP, T1, T2** (振幅、位相、Sliced Amplitude, SlicedPhase, Temp1, Temp2 を意味する)が使われる。このコマンドがあった場合、expression は最終 plot-map の値によって決まる。これは formula parser と同じである。したがって、ここに

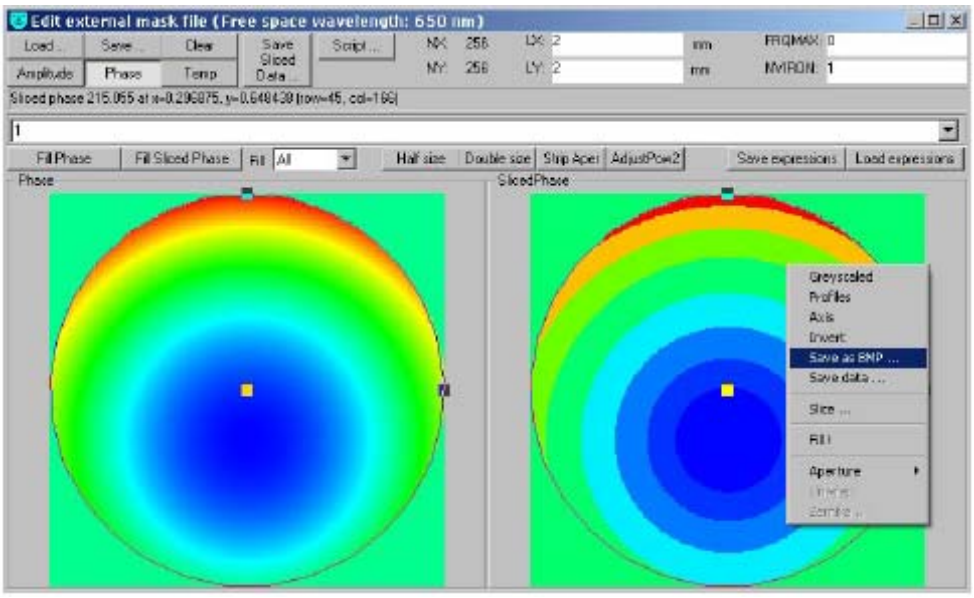
述べたすべての変数（I, J, R, X, Y, などのように）は expression の内部で使うことが出来る。これは、plot-map コマンドはここでは使われない他の命令に比してさらに一般的なものとなる。

緊急的な停止の時には、キーボードの **Escape** が Script を直ちに止めるのに使われる。

15.10 Bitmap としての単純な mask-plot の導出（Exporting Single Mask-Plots as Bitmaps）

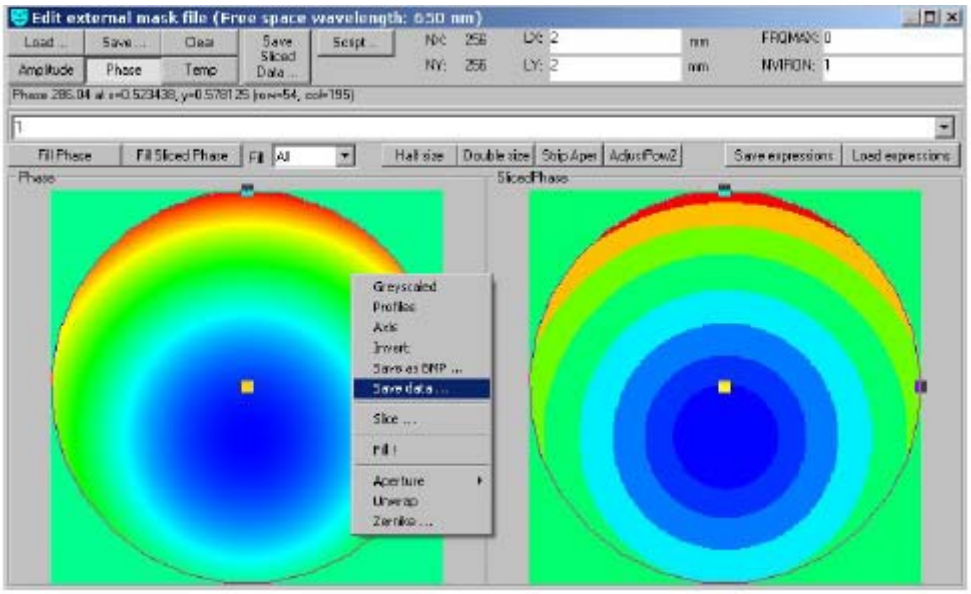
次の図に示すように、mask-Plot を bitmap として導出することは、導出しようとする曲面を表示すればよい。これには popup-menu を右マウスカーソルで開き、**Save as BMP…**を選択すればよい。

この例では、SlicedPhase plot が bitmap ファイルとして保存されている。このようにして作られた Bitmap はこの章の後で述べる bitmap importing facility で逆導入することが出来る。



15.11 ASCII データマトリクスとして単純な mask-Plot の導出（Exporting Single Mask-Plots as ASCII data-matrices）

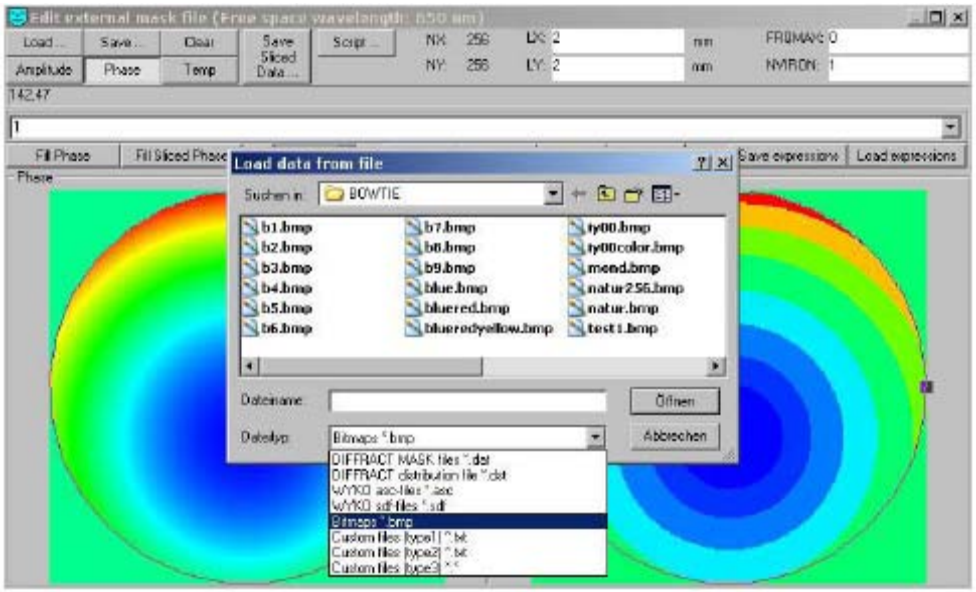
次の図に示すように、ASCII data-matrix として導出する方法は、まず導出する曲面を表示し、その後 popup menu を右マウスカーソルで開き、最後に menu から **Save data…**を選べばよい。



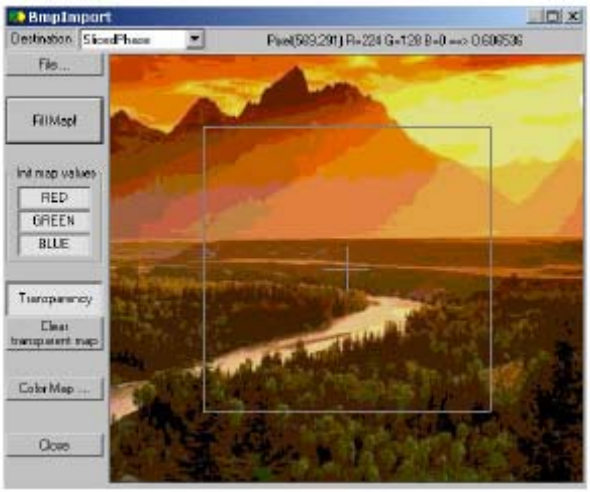
曲面の値はデータマトリックスとして、DIFFRACT の PLOT コマンドで作られたファイルと同様に保存される。他の可能性のうちで、MaskEditor でこうして作られたファイルは、後で BitmapView または ReliefViewni に導入し、ディスプレイされる。

15.12　Bitmaps からのデータの導入（Importing Data from Bitmaps）

Bitmap ファイルからデータを導入するには、MaskEditor の **Load...**ボタンをクリックし、その後 File-Open Dialog box からファイルタイプ“Bitmaps *.bmp”を選択すればよい。下図にその様子を示す。



希望するファイル名を選び、**Open** ボタンをクリックする。こうすると次の図に示すように **Bmp_Import** dialog box が開かれる。、



この時点で、bitmap ファイルは DIALOG BOX のみに読み込まれ（さらに表示され）る。すべての MaskEdito の plot map は変化しないで残る。また、256 色（あるいはグレイレベル）の bitmap のみが読み込めるというのは重要なことである。True-color の bitmap は自動的に 256 色の bitmap に変換され、色は多少変化し、bitmap ファイルの内容が変化する。

その構造は bitmap のカラーパレットにおける各色の数値を持っていて、**Fill Map** ボタンを押すと、!この値を（各色の bitmap pixel に対し）最終面に送る。これらの色値ペアーは **Color Map** に保存される。これらの色値は個々に bitmap パレットで変更することができるが、色は変更することができない。

Bitmap は左マウスボタンでドラッグすると スクリーン上で移動させることができ、マウスホイールを回転するとディスプレイされた面積を拡大または縮小できる。

Bmp_Import dialog box は次のコマンドボタンを用意している。

Destination: Combo box から bitmap data で満たしたい特定の図表を選ぶ（一時には 1 つの図表のみが選択できる)。

File...: 現在導入されている bitmap を外部ファイルで別の bitmap と入れ替える。

Fill Map!　このボタンを押すと選択した最終 plot を、最終 plot として存在しているどのようなデータで満たすことができる。この操作はやり直しが効かない。

Input map values グループの **RED, GREEN, BLUE** チェックボタンは、それぞれの色が数値がどのようにして計算されたかを決定する。ある色が赤、緑、および青で構成されている（それぞれの領域は 0 から 255 まで）として、次の式が使われる。

$$value = \frac{3 \cdot rflag + 2 \cdot gflag \cdot green + bflag \cdot blue}{3 \cdot rflag \cdot 255 + 2 \cdot gflag \cdot 255 + bflag \cdot 255}$$、

ここに、*rflag, gflag*, および *bflag* は 0 または 1 であって、RED, GREEN, BLUE ボタンがチェックされているとき（flag＝0）あるいはチェックされていないとき（flag=1）である。その結果、値は 0 と 1 の間である。

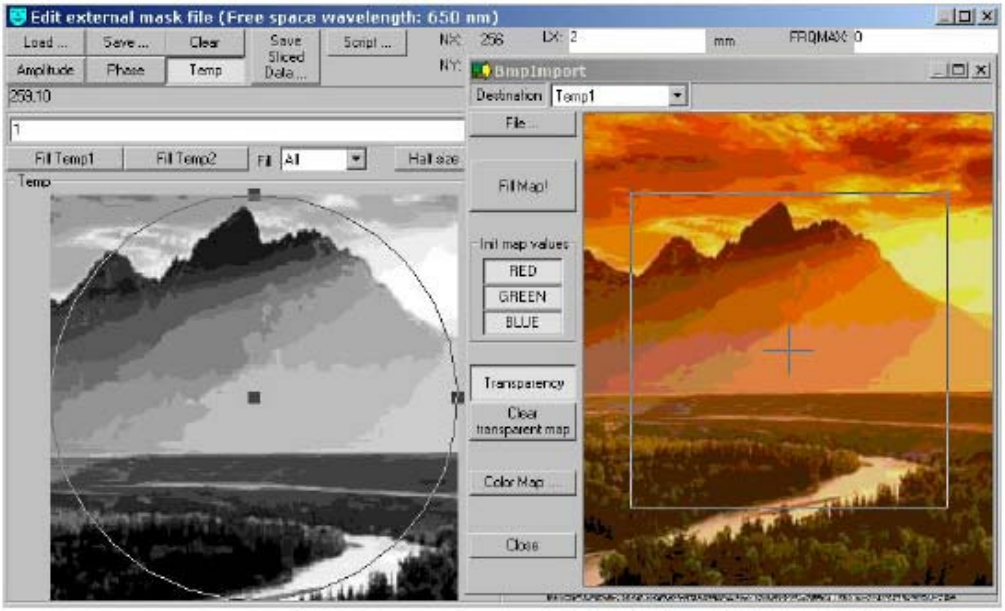
もし、RED, GREEN あるいは BLUE ボタンのどれもがチェックされていないときは、DiffractGUI は rainbow-color シリーズの値を決めるが、これは DiffractGUI から導出されるすべての bitmaps に使われている色として決められる。R, G. B の値で与えられるすべての色はレインボウスペクトラムの一部ではない。マッピング演算はこれらの色に最も近い色を探すが、この方法は結果として同じ値を持つ多数の色を導くことが多い。裁量の結果としては、この方法は DiffractGUI の BitmapView から導入された bitmaps に対してのみ使うべきである。

Transparency:　このボタンを可能にしていけば、transparent とマークされたすべての色（詳細は下に述べる）が作図から除外され、次の **Fill Map!**コマンドは tranparent color に対応する場所において destination plot に変化を与えない。もし使えなくしておけば、各色の transparent property は考慮されなくて、bitmap は全面的に表示される。

Clear transparent map: すべての色から transparent property を取り除く。その結果見える効果は **Transparency** チェックボタンで得られるのと同じである。すなわち bitmap を不透明に表示するが、重要な違いはこの場合は transparent のフラッグが永久的に取り除かれ、この場合は tranparency はそれぞれの色に対してはっきりと設定しなければならない。

Color Map…: このボタンは関連のある value-color pairs を含む小さな dialog box を開く。この box の中では map value は変化を受け、色は transparent か否かを決めなければならない。

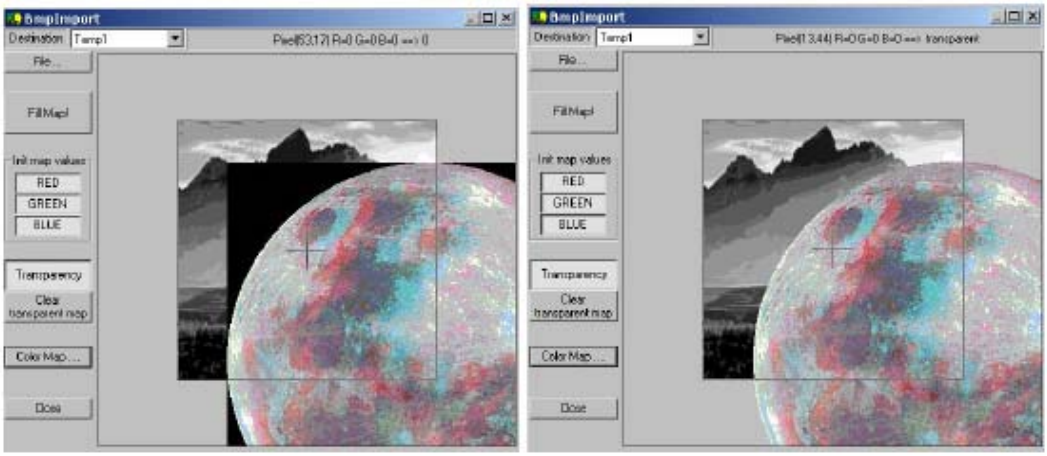
次の図は Temp1 を最終と選んだ場合 **Fill Map!**の結果を示す。



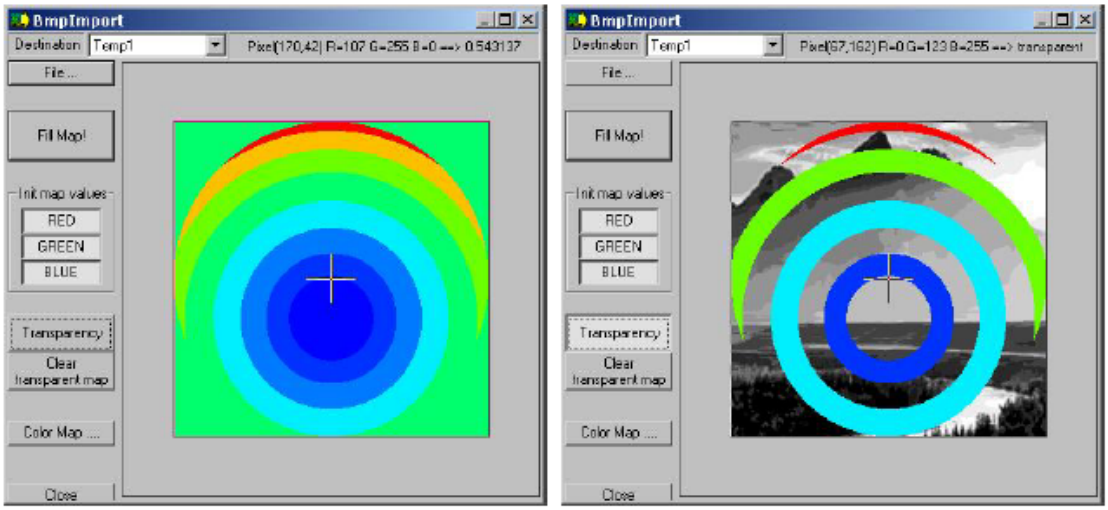
Destination plot は Bmp_Import dialog box のディスプレイ領域に矩形状にスケッチされている。その中心は十字マークで表示されていて、移動することはできないが、常に表示画面の中心に現れる。これに対し、導入された bitmap はスクリーンの周りに左マウスボタンで移動することができる。

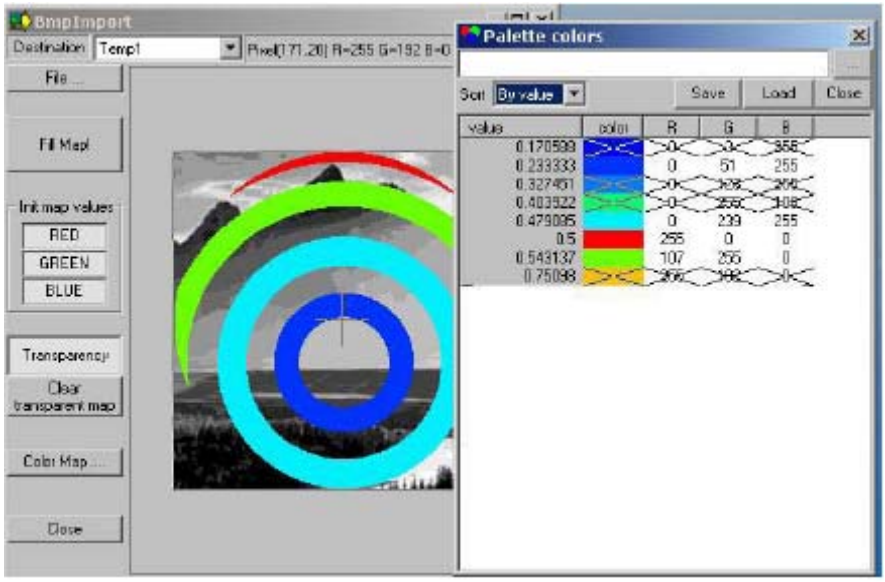
もともと、destination plot と bitmao の中心は一致している。（左マウスボタンをダブルクリックすれば、bitmap をどのように移動した後でもこの一致を回復することができる。）1 対 1 の pixel-to-cell 移動（bitmap から destination plot へ）のみが可能である。換言すれば、NMAX 行と NMAY 列をもつ局面は最小 NMAY×NMAX の bitmap のみで満たすことができる。それぞれの pixel は曲面の 1 cell に移され、伸ばすことはできない。bitmap でカバーできなかった（あるいは透明なカラーでカバーされた）plot cell は、変わらないで残される。

次の 2 つの図は color transparent を作る効果を説明している。この例では、透明になった導入された bitmap は、黒い背景色であった。



Ctrl ボタンを押し下げ左マウスボタンを bitmap 上でクリックすれば、マウスカーソルの下の色の transparency は切り替わる。これは visible color のそれぞれの transparency を決めるための手早い方法である。もし、transparency オプションが可能であれば、これが直ちにスクリーン上で反射する。次の２つの図はこの効果を示している。





Color Map…ボタンで Palette Colors dialog box を開くことにより、その色につけられていた値を（手動で）変えることができる。Value-color pair は外部テキストファイルに記憶され、後で **Save and Load** ボタンで読み込むことができる。しかし、記憶された value-color map は正確に同じ color palette を持つ bitmap にのみ有用である。したがって、（自動的に framework で作られた）初期値から変化している数値が入用なときは、それぞれの使用した bitmap について、bitmap の color-palette にマッチする value-color pair を記憶することが必要となる。

（グレイの背景で表示された）リストの数値を左マウスでクリックすると、数値を編集する edit box を開くことができる。リストの中にある色のついた矩形をクリックすると、transparency flag が切り替わる。transparent color に登録すると、それらは bitmap では見えないと表示される。登録は未定義の順序または種々のパラメーターでソートされて表示される。表示の順序は bitmap の表示または導入自身にまったく関係しないことを注意されたい。

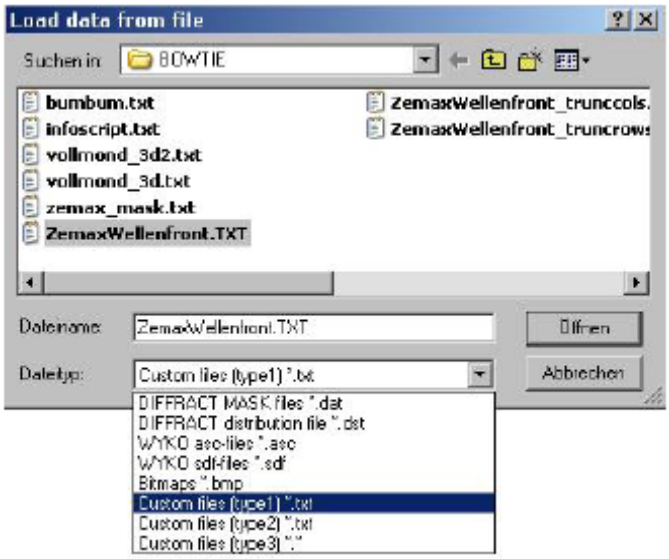
最後に、Palette にあるすべての色は bitmap にも存在しているわけではないが、bitmap で使われている色はすべて color palette に存在していることに注意すべきである。どの場合にも、DiffractGUI はすべての色について数値を指定する。

15.13 任意の ASCII ファイルからのデータ導入（Importing Data from Custom ASCII Files）

、

この特性は、ユーザーに external mark や distribution files が DIFFRACT の仕様と一致しない他のプログラムで作った ASCII ファイルを導入することを可能とする。もっとも重要な要求は、(義務的な) データと (任意の) ヘッダーがデータファイルの中ではっきりしたブロックを構成していることである (これが通常の場合である)。データは cell の中でグループとなり、それぞれの cell は単純な数値または (実数と虚数) の組み合わせによる複素数であることが望ましい。実際には種々の異なった形式が存在するので、DiffractGUI が特定のファイルからデータを導入する前に、何らかの"training"が必要と思われる。　これは次の例で明らかになるであろう。

スタートするには、MaskEditor の Open…ボタンをクリックし、File-Open dialog box で下図に示すように、3 つの"Custom files"が現れるので、そのひとつをから選択する。



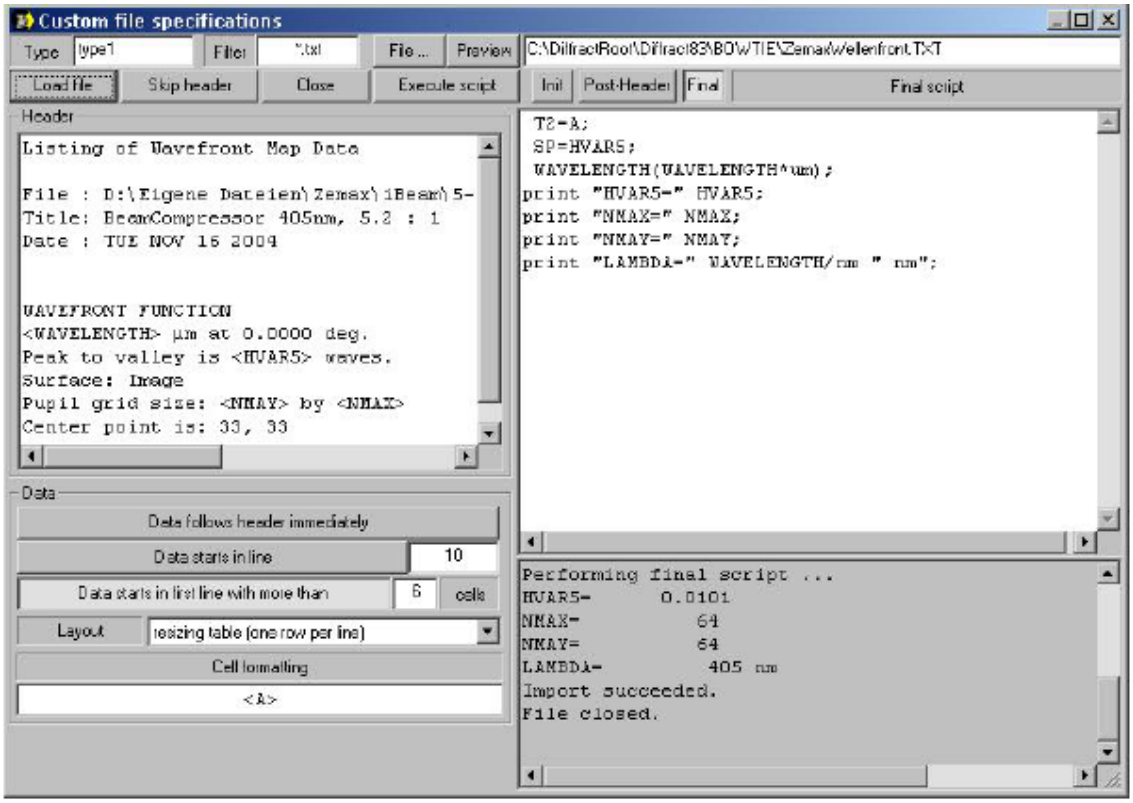
3 つの"Custom files"は同じ関数姓を持つ；後でそれぞれは異なったファイルタイプを読むようにプログラムされているので、多数の読み込みを設けるのが便利である。ファイルタイプを変えるたびに、導入フィルターを再プログラムする必要がある。

この段階で、どのファイルを選んだかということは重要ではない。続けるには、どれかのファイルを選び、または edit-box にはいっているどれかの字をタイプして、**Open** を押す。新しい dialog box, **Custom_file_specifications** が下図のように開かれる。、

一般に、導入過程は次のような順序で行われる。

1. Init-script を走らせる。
2. ファイルヘッダーを解析する。
3. Post-Header script を走らせる。
4. データ値を計算する。
5. Final-script を走らせる。

ユーザーは **Custom_file_specifications** dialog を使ってこれらのステップを個々に編集することができる。下図はこのことを例として示す。



“Type”および”Filter”のラベルは任意に変えることができ、それらは次の実施のとき **File-Open** dialog box に現れる。それに加えて、Windows で通常使われるように、”filter”は特定のファイルのみを決められた拡張子をつけて表示する。特に他に特別なものは何もない。

File…ボタンは、MaskEditor に戻ることなく、データを導入する他のファイルを選択することができる。ファイル名は直接 edit box で変更することができる。Preview ボタンは小さい text-viewer を開き、選択したファイルの内容を表示する。これはデータの導入の準備のとき便利である。（あるいは、トラブルが発生したとき、導入しようとするデータの仕様

が適応しているかを知り、デバッグのために便利である。）

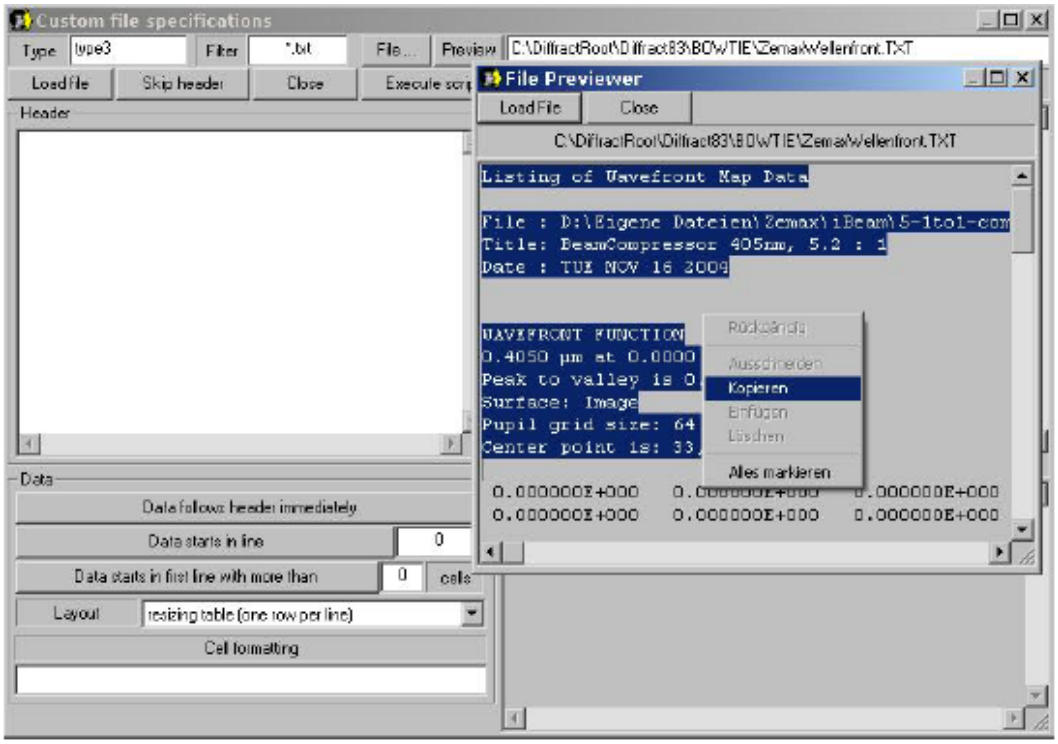
データ導入のとき使われる 3 つの script があり、これら 3 つのすべてを予知できるただ 1 つの edit box である。すなわち **Init, Post-Header** および **Final** のチェックボタンはをこれらのひとつを見たり編集するために使われる。**Execute_script** は、全体の導入プロセスを実行しないで、選択した script を直ちに実行できる。 Preview オプションと同様に、いくらか制限はあるが、これは準備あるいは debugging のときに有用である。

通常、導入機能のプログラミングは次のオーダーで起こる。最初に header と data format を決める。これは、もし必要なら、scripts を書く後で起こる。

例 1

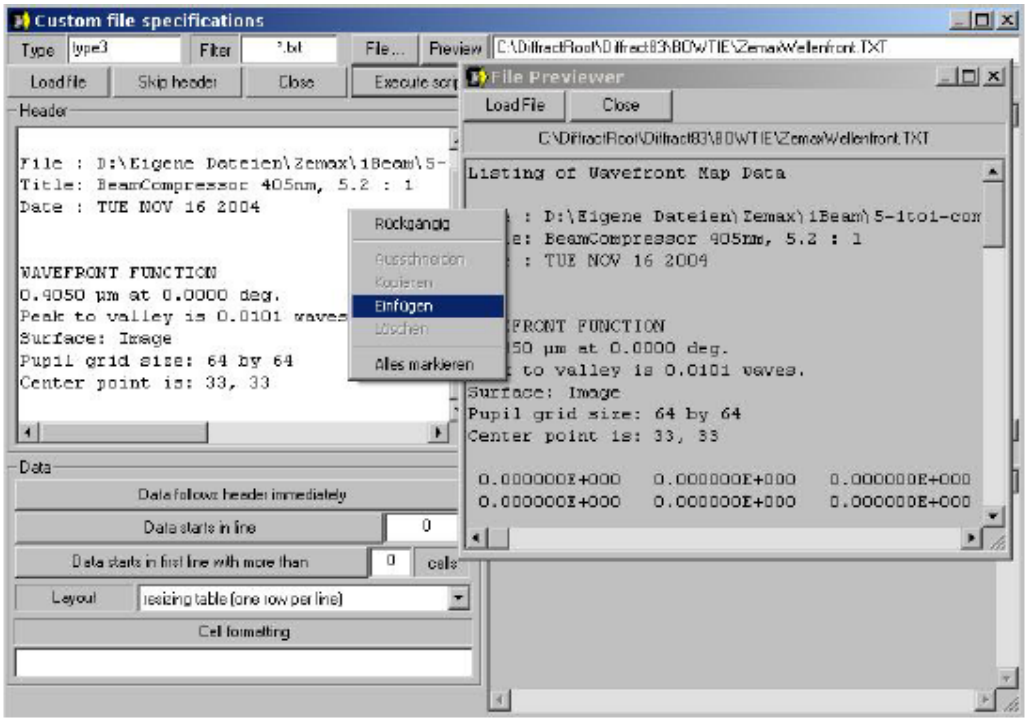
この例では、Zemax™ プログラムで作られた波面マップを読み込むプログラムを作る。

最初に、**Preview** をクリックし、complete header area を選ぶ。右マウスボタンをクリックし **Copy** を選択し、下に示すように、それをクリップボードにおく。



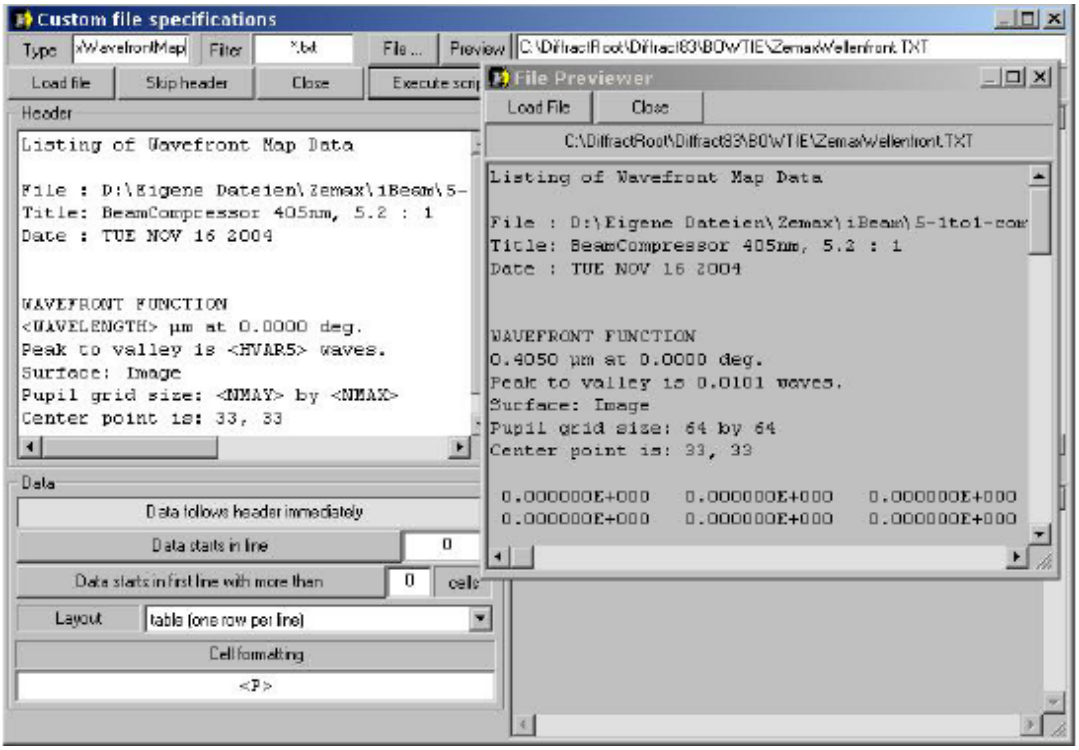
ここで、（空の）”Header” edit box をクリックし、次の図に示すように、クリップボードか

ら前に保存した header を貼り付ける。この header はファイルから特定の数値を抽出する NMAX あるいは NMAY のテンプレートの役割をする。これらの数値のそれぞれは導入のとき働く script のひとつとして使われる変数に保存される。Framework は次の変数を与える： NMAX, NMAY, LMAX, LMAY, WAVELENGTH, HVAR1, HVAR2, HVAR3, HVAR4 および HVAR5。変数 HVAR1 から HVAR5 は一般的用途であって、他の特定の値を保存するのに使われる。



次のステップでは、header からスキャンされるそれぞれの数値は先に述べた変数名に置き換えられなければならない。次の図に示すようにそれらは＜と＞で囲まれ、NMAX, NMAY, および WAVELENGTH の位置は、<NMAX>, <NMAY>, および<WAVELENGTH>の数値で置き換えられて、マークされる。

、



それに加えて、“Peak to valley”値はデモンストレーション用に<HVAR5>に読み込まれる。**Skip header** チェックボタンは header がデータファイルに存在するかどうかを決めるのに使われる。上の例では header が存在するので、このボタンはチェックされないで残る。(もし、Skip header ボタンがチェックされると、その効果は header template box は空のまま残るのと同じである。)

それぞれの<variable>entry の前後には leading および trailing スペースを置くことは重要なことである。これらのブランクは DiffractGUI に導入したファイルの item の間に適当なブランクを受け入れることを教えている。このようにして、フォーマットの柔軟性を増している。

この点において header の準備は終了し、データ形式は特定される。最初に、ファイルの内部でどの位置からデータ領域が始まるかを指定しなければならない。次のオプションが可能である。

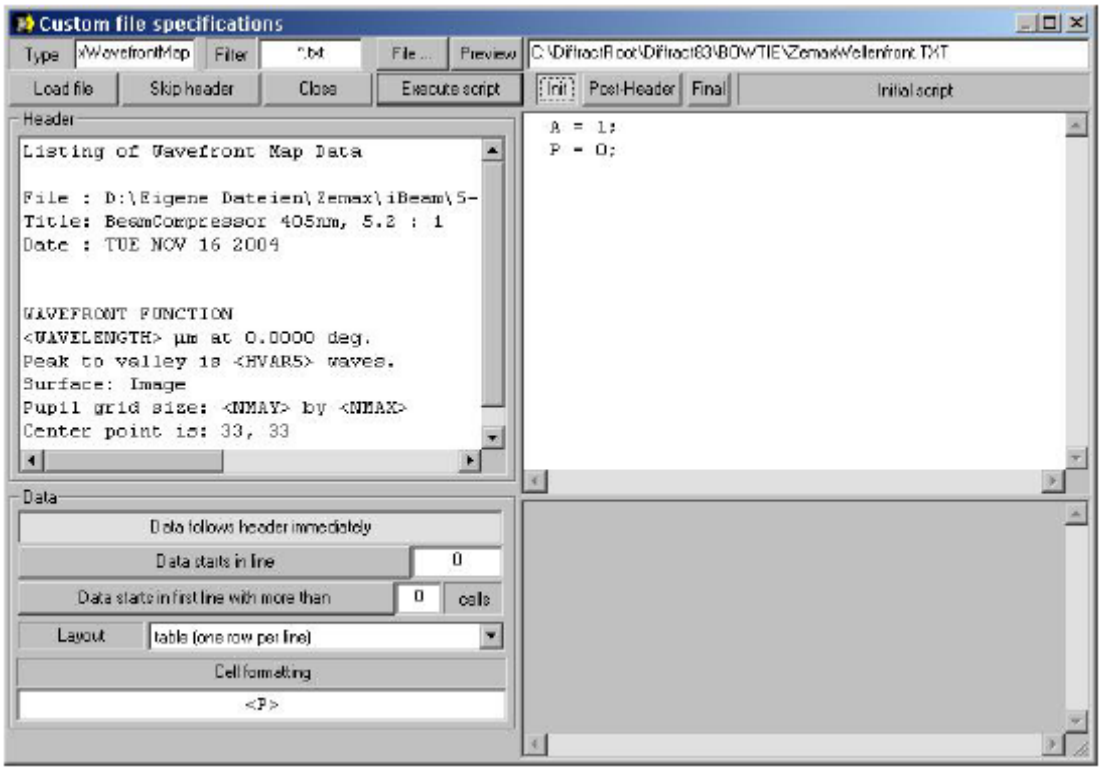
- ある行からデータが始まる
- データは明確な数の cell を持つ最初の行から始まる
- データは header から直ちに始まる

行番号は 0 から始まる。行を数えるとき空の行（あるいはブランクだけの行）は自動的に除外されことを記憶しなければならない。したがって、これらはデータ導入の過程で数えられない。Preview から、データブロックは header の直後から始まので、次の図に示すように、対応するボタンはチェックしなければならない。

データブロックが確認された後には、データレイアウトを選ばなければならない。これには 4 つのレイアウトが可能である。

- 表
- リサイズした表
- 列（行状）
- 列（列状）

表とリサイズした表は基本的には同じものである。唯一の違いはリサイズした表は自動的にすべての mask plot をファイルの行と列の数に正確に一致させているが、表レイアウトはデータを常に（行＝0、列＝0）からスタートしサイズの変更はしない。もし、destination plot がデータブロックより多い行や列を持っているときは、それらは部分的にしか取り入れられない。逆に、データーブロックにある余分な行や列は無視される。



表とリサイズした表の両者は、データが次のように配列されるのを期待している。

Cell 0,0	Cell 0,1	Cell 0,2
Cell 1,0	Cell 1,1	Cell 1,2
Cell 2,0	Cell 2,1	Cell 2,2
・・・	・・・	・・・

それぞれの行は正確に 1 行を含まなければならない。これに対し、表（行状）レイアウトはデータがまた行によって順序つけられていると考えられている。しかしファイルの行はマトリックスの全列を含む必要はない。特別な場合、行あたり 1 数値、たとえば

Cell 0,0
Cell 0,1
Cell 0,2
✧ ・・・
✧ Cell 0, NMAX-1
✧ Cell 1,0
✧ Cell 1,1
✧ ・・・
✧ ・・・
✧ Cell NMAY-1, NMAX-1

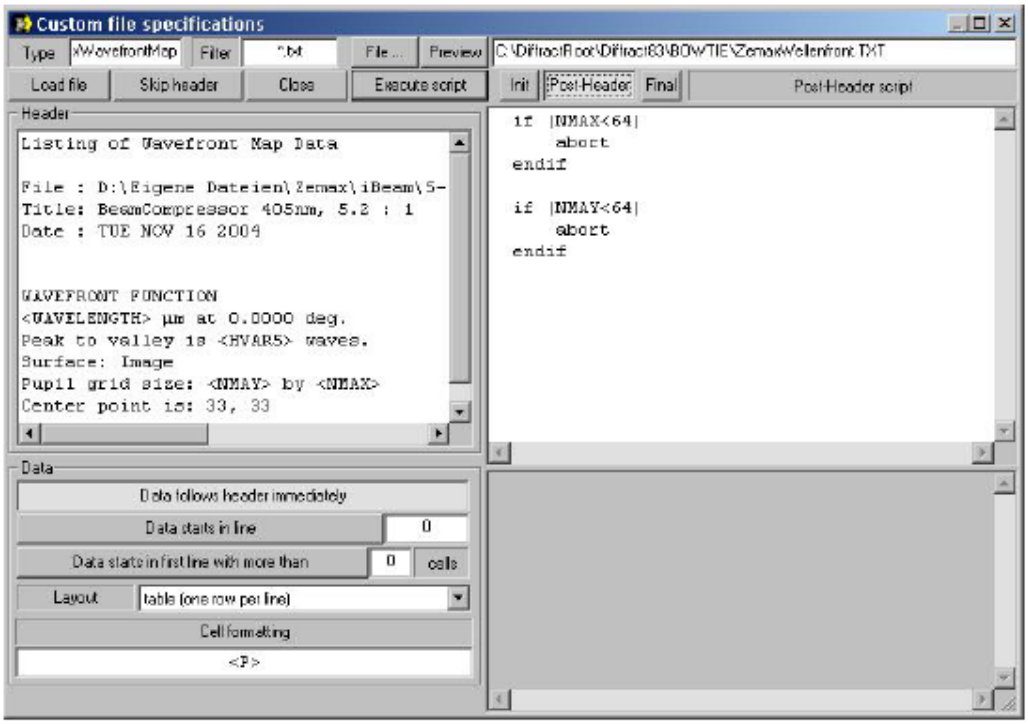
名前が示すように、列（列状）レイアウトは列（行状）と同じようであるが、データが列状に配列されることを期待されている。

Cell 0,0
Cell 1,0
Cell 2,0
✧ ・・
✧ Cell NMAY-1,0
✧ Cell 0,1
✧ Cell 1,1
✧ ・・・
✧ Cell NMAY-1, NMAX-1

選んだ例では、Preview で見られるように、表あるいはリサイズした表のレイアウトは適合している。

最後に、データ仕様のために、cell の配列を宣言しなければならない。この例では、それぞれの cell は単一のデータ値で構成されている。**Cell formatting** edit box は cell レイアウトを識別するのに使うことができ、さらに重要なことは MaskEditor の 6 種の可能な plot のうち、どれがデータを記録すべきかを決められる。ブランクで分離された単一 cell（この例

の場合であるが）に対しては、導入のときデータを Amplitude, Phase, SlicedAmplitude, SliedPhase, Temp1,あるいはTemp2のフレームに送りたいときは、それぞれ単に<A>, <P>, <SA>, <SP>, <T1>, あるいは<T2>とタイプすればよい。 次の図は Phase plot をデータで満たしたいときの例である。



・

ブランク（空の空間）を cell format text（図では見えない）の前と後ろに置くことは重要なことである。分離のための文字もこの box で指定することができる。たとえば、もしデータ値が”,”（コンマ）で分離され、ブランクでない場合（この例がそれである）、＜P>の代わりに

< P > ,

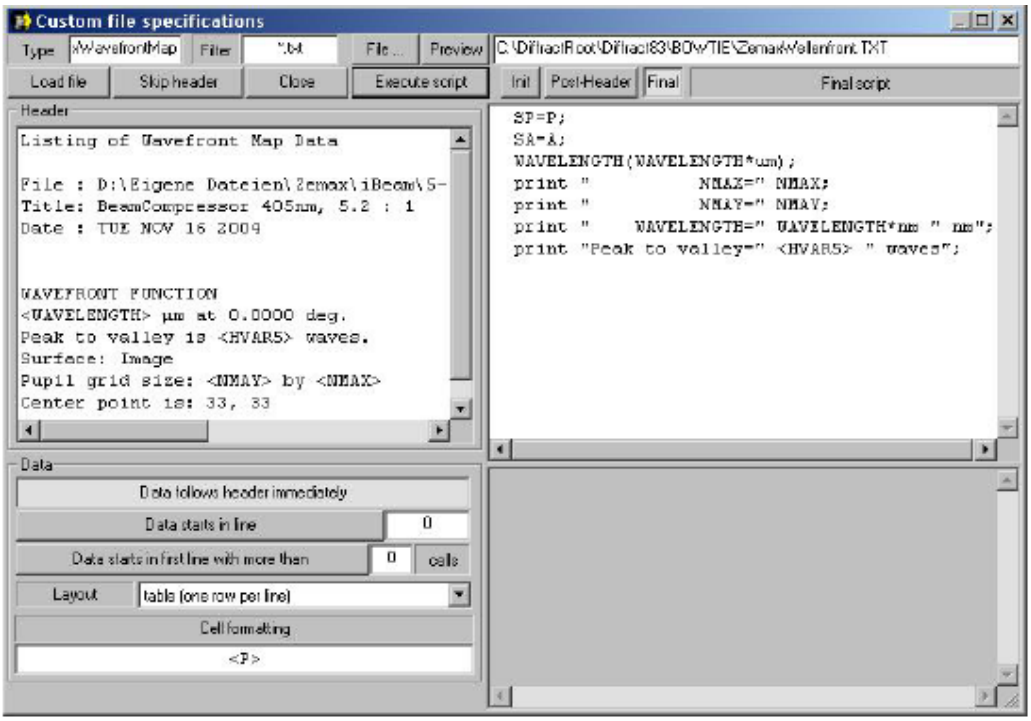
で指定する。”>”と”,”の間のスペースに注意されたい。次の例はさらに複雑な cell の cell-formatting を示す。

ファイルを導入する前の最後の準備は、3 つの scripts を書くことである。それらを使うことは必須ではない；もし script を使うことを希望しないときは、script-text は空白で残すか、あるいは最初の script command として”stop”を使うことである。

Init-script は曲面を初期化するのに最良の場所である。この例では

```
A = 1;
P = 0;
```

これは Amplitude plot map は 1 で満たし、Phase plot map は 0 で満たすことを意味している。

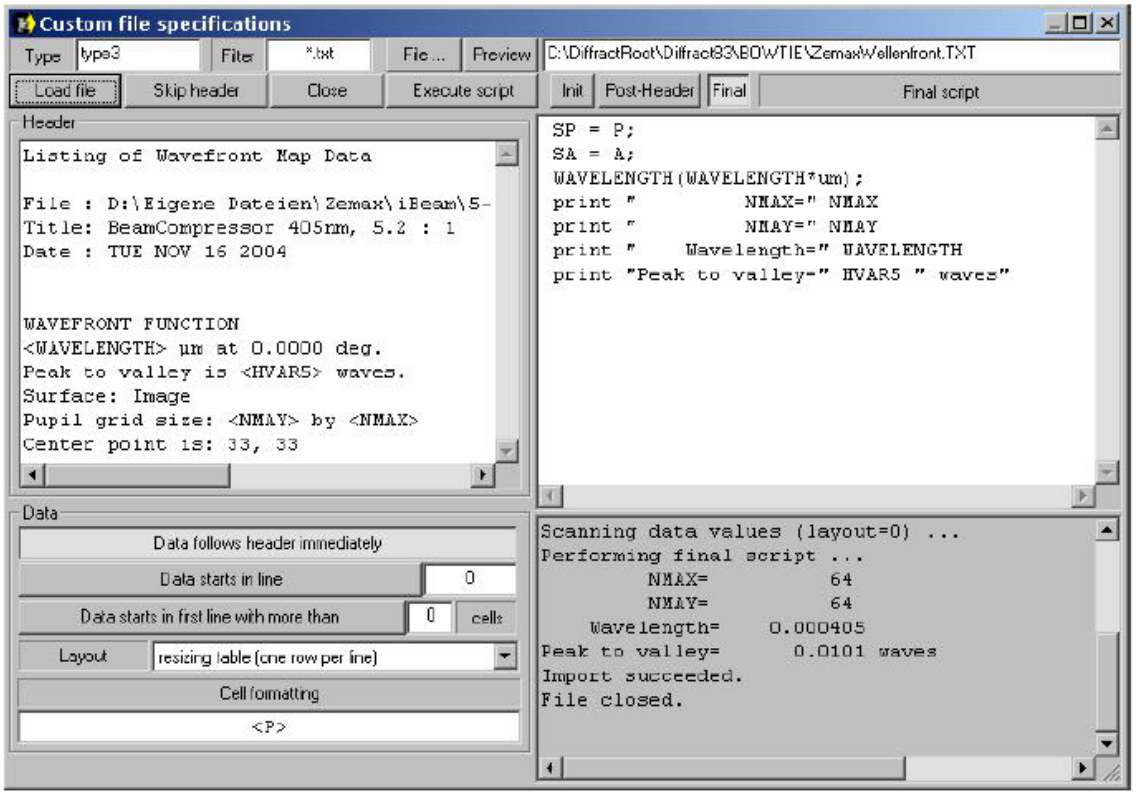


Post-Header script は header から導入したデータを保障するのに最良である。この例では、NMAX と NMAY が 64 より小さいことを保障する。

```
if (NMAX<64)
     abort
endif
if (NMAY<64)
     abort
endif
```

Script の実行中、エラーが発生した場合、導入プロセスは停止する。したがって、導入プロセスを故意に停止させたいとき、何か意味のない script（上の例では、知られていないコマンド”abort”）を書くとよい。故意の syntax error は自動的にプロセスを停止させる。

Script の中で新しく定義された変数（”var”コマンド）はそれ以後の script までパスする。たとえば、Inst-Script で定義されたすべての変数は Post-Header と Final-Script でも使用できるが、Final-Script で定義された変数は Init または Post-Header Script では使えない。



Final script は closing activity で使用できる。たとえば

```
SP = P;
SA = A;
WAVELENGTH(WAVELENGTH*um);
print "       NMAX=" NMAX
print "       NMAY=" NMAY
print "  WAVELENGTH=" WAVELENGTH
print "  Peak to valley=" HVAR5 " waves"
```

は Phase map を SlicedPhase map にコピーし、Amplitude map を SlicedAmplitude map にコピーし、さらに変数のいくつかの値をプリントする。

準備が終わると、導入プロセスをスタートするために **Load file** をクリックする。

Framework は、上の図に示すように、dialog box の右下にある（灰色の）edit box に import-status を知らせる。全体の出力は下にリストを載せた。

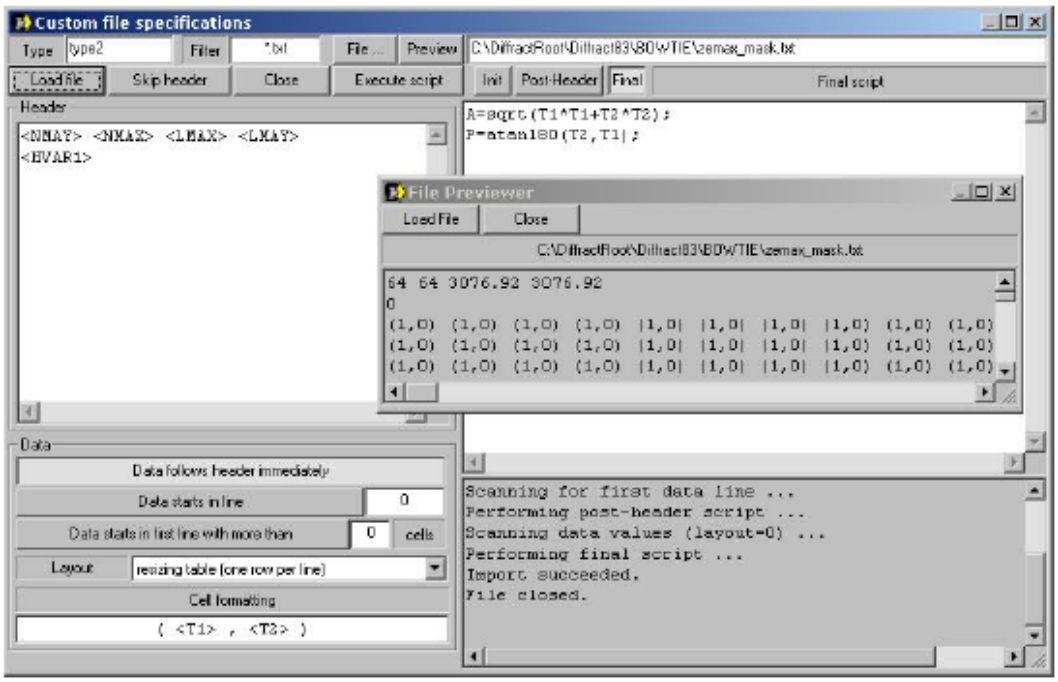
便宜上、File-header の走査中に出会ったそれぞれの変数は、その場所の header にある数値とともにプリントされている。

```
Start loading
C:\DiffractRoot\Diffract83\BOWTIE\ZemaxWellenfront.TXT
Performing initial script ...
Scanning header ...
WAVELENGTH(0.405);
HVAR5(0.0101);
NMAY(64);NMAX(64);
Scanning for first data line ...
Performing post-header script ...
Scanning data values (layout=0) ...
Performing final script ...
        NMAX=         64
        NMAY=         64
   Wavelength=    0.000405
Peak to valley=      0.0101 waves
Import succeeded.
File closed.
```

Custom_file_specifications dialog box はモードレスでサイズの変更ができる。これは導入の後は MaskEditor に戻るために dialog-box を閉じなければならないことはない。しかし、MaskEditor が閉じられると、それは自動的に閉じられる。MaskEditor が閉じられると、すべての設定はレジストリーに保存され、将来のセッションのために保存される。

例２

この第２の例は、DIFFRACT の XOBJ に使うため、ASCII マスクファイルの導入を説明する。前の例との主な違いは、次の図の **File Preview** ウインドウで示したように、データセルが（real, imaginary）の複素数で書かれていることである。



Header は NMAX, NMAY, LMAX, LMAY を除き、他のアルファベット文字を含んでいないから、比較的簡単である。変数間のブランクを注意されたい。

１つの cell は２つの変数で構成されているため、データ形式は次のようになっている。

```
( <T1> , <T2> )
```

これは DiffractGUI にそれぞれの cell の最初の値を Temp1 map に、２番目の値を Temp2 map に記憶するように言っている。DiffractGUI は cell を”(“と”)”で囲み、２つの値は”,”（カンマ）で分離する。

この点を再び表示するには、たとえば、もし外部ファイルのデータ形式が、たまたま**{value; value} {value:value}…**となっていたら、cell フォーマットは次のように決めなければならない。

```
{ <T1> ; <T2> }
```

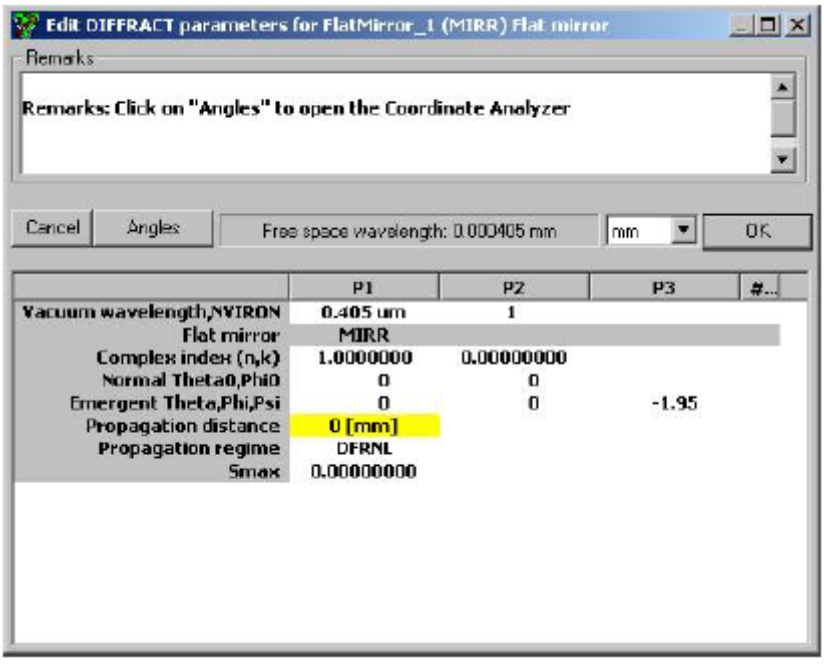
数値の間のブランクについては、DiffractGUI に数値の間のブランクを無視するように伝えることに注意されたい。一般に cell は２つ以上の数値を含み、これは cell フォーマットで規定しなければならない。しかし、データブロックのすべての cell は同じパターを持たなければならない。そうでなければ、走査は失敗する。また、cell の各値は異なった plot map へ移動することが望ましく、そうでなければ（データのいくつかは）失われる。
この例における Init-および Post-Header script は最初の例と同じであるが、Final script

は Temp1 および Temp2 に保存した複素数の実数および虚数から振幅と位相を計算するように変えてある。計算結果は Amplitude map と Phase map に保存する。

```
A = sqrt(T1*T1+T2*T2);
P = atan180(T2,T1);
```

16. 座標の解析（Coordinate Analyser）

Coordinate Analyser は Euler 角（DIFFRACT の取扱説明書で Euler 角の詳細な記述を参照のこと）で記述された種々の object の空間的な方向を眺めるためのグラフィックウインドウである。いくつかの DIFFRACT の Command（MIRR, MLYR など）は、入射光、射出光、および object そのものの方向は (θ, ϕ) または (θ, ϕ, ψ) の角を必要とする。Coordinate Analyser は、MIRR あるいは MLYR と一緒に起動され、決めた Euler 角が用途に合致しているかを確認するのに役立つ。



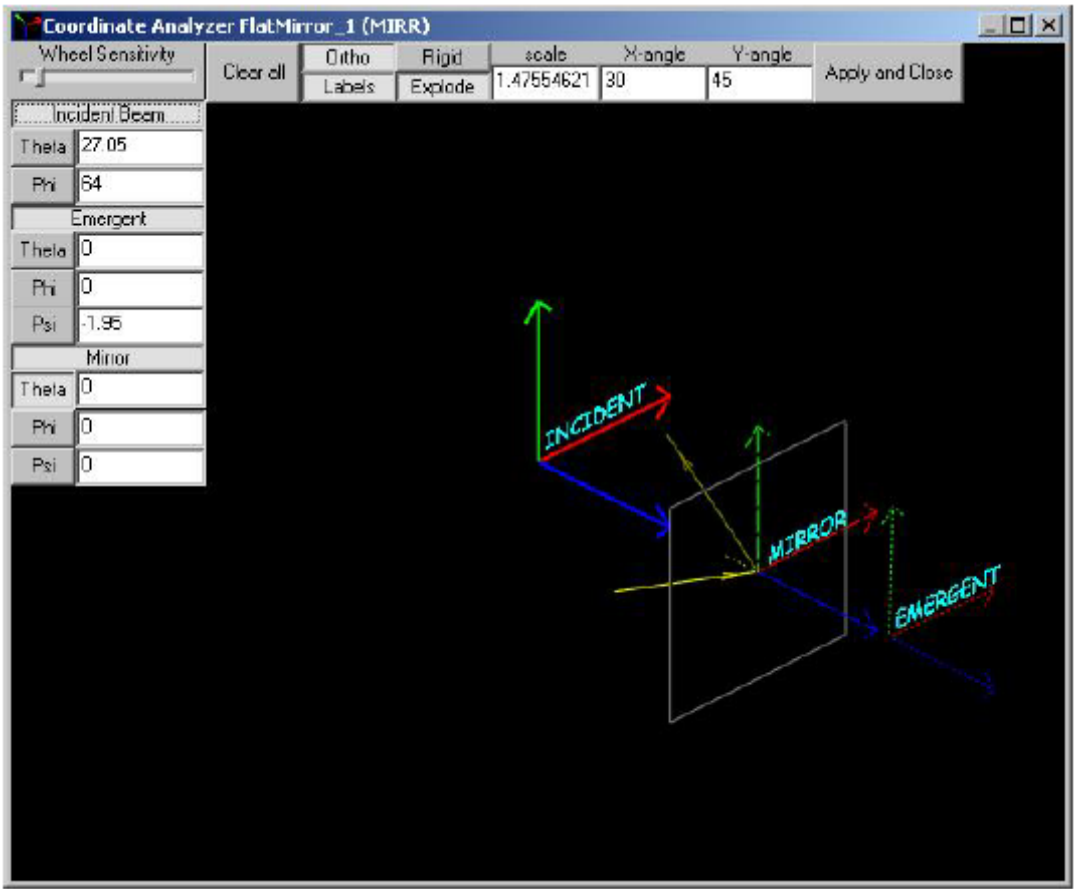
上図は **Coordinate Analyser** が Edit_DIFFRACT_parameter 対話ボックスからどのようにして起動されるかを示す。Coordinate Analyser ウインドウを開くには、**Angles** ボタンをクリックする。問題の角は Coordinate Analyser に移り、そこで種々の方法（キーボードまたはマウスホイール）で変化させることができる。これらの変化した角度は、Coordinate Analyser の閉じ方に従って、command block に読み込まれたり廃棄されたりする。**Coordinate Analyser** はこのようにして、選んだ角度が目的としている応用に適当であるか

どうか確かめて、ユーザーに DIFFRACT の Euler 角に慣れさせるのに使われる。**Coordinate Analyser** の種々の機能については、次の節で詳しく述べる。

16.1 Mirror

次の図は **CoordinateAnalyser** を Mirror(MIRR)object に対して起動したものである。ウインドウに表示されている種々のコントロールボタンの詳しい説明をする前に、これらのコントロールは次の２つのカテゴリーにグループ分けされていることを述べる。

- 表示だけを変化させるもの。これらのほとんどはウインドウの上部の水平バーの中にある。
- DIFFRACT コマンドのパラメーターを変化させるもの。これらはウインドウの左側の縁に垂直に配置されている。



CoordinateAnalyser は INCIDENT, MIRROR, および EMERGENT とラベルされた 3 つの直交座標軸を表示している。 x 軸は赤、 y 軸は緑、 z 軸は青である。これら 3 つの座標軸は異なった線の種類および異なった長さで表示されているので、たとえ重なっても区別

できる。**Incident Beam** は黄色の矢印で表示され、（計算された）**Emergent Beam** は茶色の矢印で表示されている。**Mirror object** 自身はグレイの矩形枠で MIRROR の 3 軸*に密接して置かれている。

さらに見やすくするために、入射ビーム（すなわち黄色の光線)、ミラーの 3 軸、射出ビームの 3 軸はそれぞれ独立に **Incident Beam, Emergent, Mirror** ボタンをチェック／アンチェックして取り除くことができる。INCIDENT の 3 軸は、すべての表示成分の基準システムとなっているため、常に表示されていて取り除くことはできない。シーン*の方向は **X-angle** および **Y-angle** の値を変える（その後 Enter を押す）ことによって変更することができる。あるいは、もっと実用的にはマウスで変えることができる。すなわちマウスカーソルを暗い背景の任意の位置に置き、左ボタンを押したままカーソルを動かせばよい。

INCIDENT の 3 軸の原点は通常ウインドウの中心にあるが、これはマウスで移動することができる（CTRL キーを押したまま、上に述べた dragging を行えばよい)。どの処理も、**CoordinateAnalyser** のウインドウの中心の周りに回転しているので、INCIDENT の 3 軸あるいは全体のシーンがスクリーンから外れてしまうことがある。左マウスボタンをダブルクリックすると、INCIDENT の 3 軸の中心からの移動はすべてキャンセルされ、その原点がウインドウの中心に戻る。

シーン全体の大きさは **scale** 値を変えることによって設定でき、*Enter* キーを押すか、マウスホイールによって実行できる。

マウスホイールの使用（Using the mouse wheel）

CoordinateAnalyser 内では種々の方向角はマウスホイールで変更することができる。マウスは 1 つのホイールしか持っていないので、角の選択はウインドウの edit control の **Theta**（θ）, **Phi**(ϕ), **Psi**(ψ)ボタンで行う。ある特定の値を変えるときは、それに対する **Theta, Phi,** または **Psi** ボタンを選び、マウスのホイールを廻せばよい。一度には一つの値しか変えられないので、１つのボタンをチェックすれば、他のボタンは自動的にチェックが外れる。

他の方法として、変更を希望する角を 3 軸上で左マウスボタンをクリックしてもよい。クリックした軸に応じて、次の値が選択できる。

MIRROR z 軸：Mirror Thata

---

* x, y, z 直交座標軸

* scene：シーンと訳した。モニターディスプレイに現れた目的とする映像を指す。

MIRROR x 軸：Mirror Phi
MIRROR y 軸：Mirror Psi

EMERGENT z 軸：Emergent Theta
EMERGENT x 軸：Emergent Phi
EMERGENT y 軸：Emergent Psi

Yellow incident beam：Incidnet Beam's Theta
Brown emergent beam：Incident Beam's Phi

もしマウスホイールにどの角も割り当てなかった場合（すなわち、すべての選択ボタンをチェックしなかった場合)、ホイールを回転すると scale 値が変化する。もちろん、マウスにホイールが無いときは、上記の機能は使うことができないので、すべてのパラメーターを希望の値にタイプし、*Enter* を押せばよい。マウスホイールから受けるインパルスのステップサイズはウインドウの左上の隅にある **Wheel Sensitivity** スライダーで調節できる。**Cear All** ボタンはすべての角をゼロにセットする。

Display Option

シーンは次のような option で変更できる。これらはすべて独立に設定できる。

- **Ortho** はシーンを正射投影†に設定する。もし不可能なときは、さらに現実的な遠近投影*を用いる。これら 2 つの投影モードの唯一の違いは、正射投影では深さの情報は考慮されていないことである。言い換えれば、正射投影では object のサイズはカメラがどれだけ遠いかまたは近いかに依存しない；つまり常に同じサイズである。
- **Labels** は 3 軸に付けられた INCIDENT, MIRROR および EMERGENT のラベルを隠すか表示するかを決める。
- **Explode** は MIRROR と EMERGENT の 3 軸の原点位置を見やすいように分散配置する。もしこの option をチェックしないと、すべての 3 軸は共通の原点（通常はシーンの中心）で描画される。
- **Rigid**.　この option では、EMERGENT の 3 軸は（計算された）射出ビームに固定され、その方向は手動で変更できない。しかし、入射ビームの方向を変えると、射出（反射）ビームもそれに従って変化し、その点で射出の 3 軸も自動的に射出ビームに従う。

Coordinate Analyser を閉じる（Closing the CoordinateAnalyser）

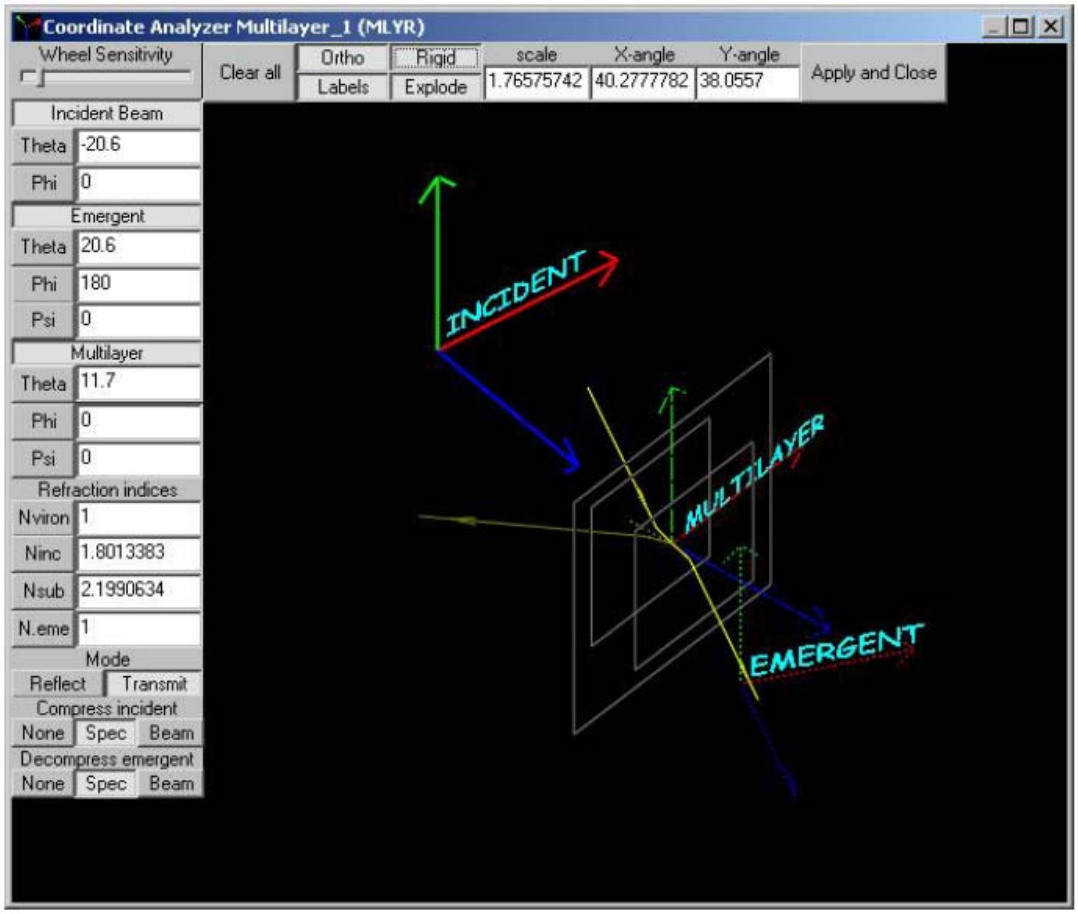
† orthographic projection：正射投影と訳した。
* perspective projection：遠近投影と訳した。

CoordinateAnalyser を閉じるには、2 つの方法がある。

- （他のウインドウを閉じる時と同じように）ウインドウの右上の隅にある×ボタンを使う。セッションで行った変更はすべて廃棄され、control は（変化なしの）**Edit DIFFRACT parameters** 対話ボックスに戻る。
- **Apply and Close** ボタンを使う。この場合は、CoordinateAnalyser が閉じる前に、(Theta, Phi, Psi の)選択した値は **Edit DIFFRACT parameters** 対話ボックスに送られる。選択した値は実行中の command block に反映される。

16.2 多層膜（Multilayer）

次の図は多層膜（**Multilayer** MLYR）object に対して起動した **CoordinateAnalyser** の外観である。



基本的な機能は（Mirror のラベルが Multilayer に変わっている点を除き）**Mirror** についての前述の事項と同じであるが、DIFFRACT の MLYR command に特定された新しい

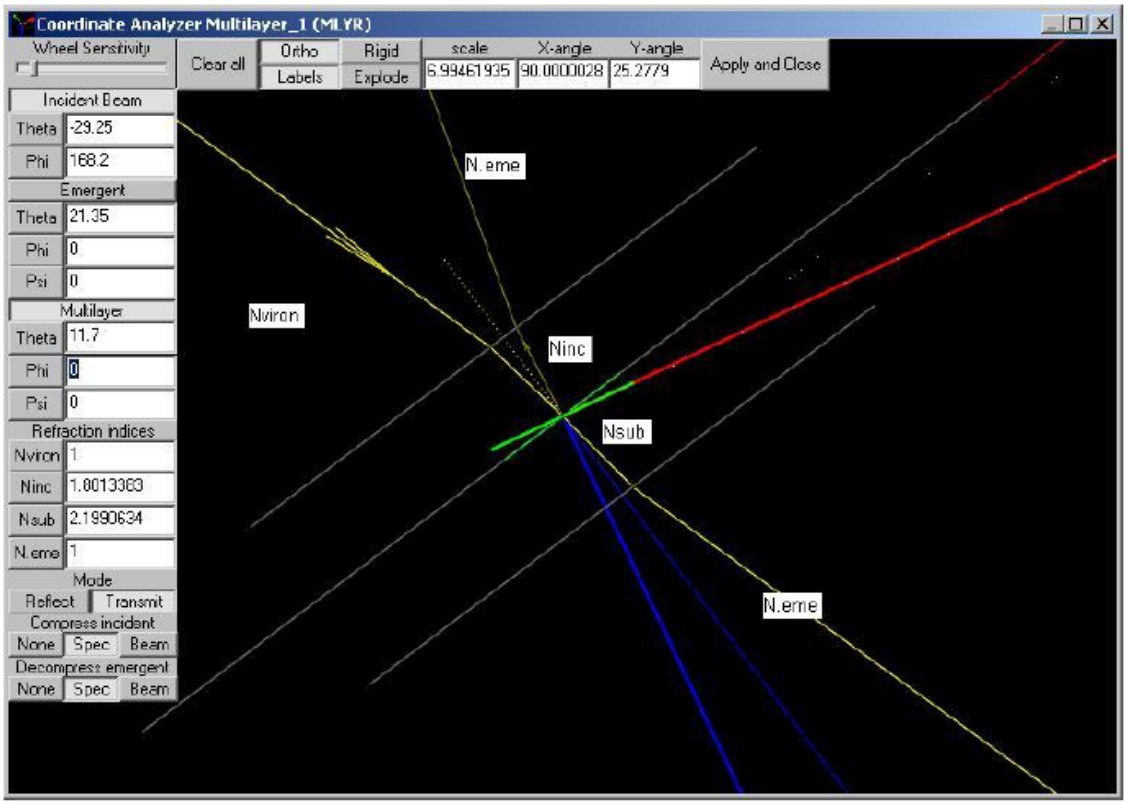
control がスクリーンに現れている。

Mirror object の場合と同じように、**Multilayer** はグレイフレームで表示されている。Compression/decompression option が spectrum に存在すれば、前の例と同じように、2 つの小さい矩形フレームが（平面）入射および射出光学面を表すのに用いられている。（これらの option については、DIFFRACT のマニュアルを参照されたい。）

黄色の矢で表された入射ビームは選択した **Mode**（すなわち **Reflect** または **Transmit**）にかかわらず、反射光線（茶色）と透過光線（黄色）に分割される。すべての光線は光学面を通過するとき Snell の法則にしたがって屈折する。

Rigid option をチェックすると、EMERGENT の 3 軸は、選択した **Mode** が **Reflect** または **Transmit** にしたがって、反射または透過光線に付けられている。

シーンにおける光線を計算するための種々の屈折率は、Euler の角（Theta, Phi, Psi）と同様に、マウスホイールまたは数値をタイプして *Enter* を押して変更することができる。次の図は、光線の通路を計算するために、どのように屈折率が使われるかを示している。



最後に、**Apply and Close** ボタンで **CoordinateAnalyser** を閉じると、すべての屈折率、

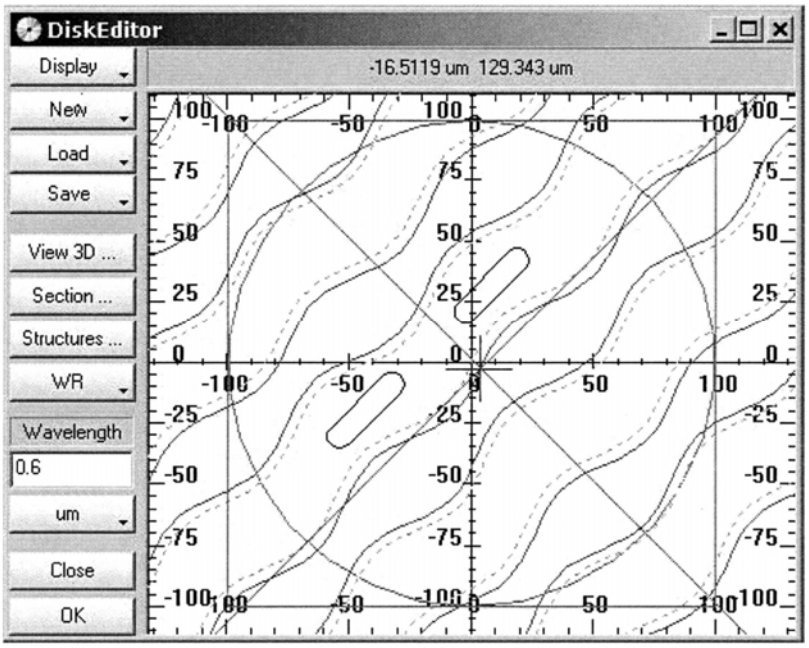
Emergent および **Multilayer** orientation angles, **Mode**, **Compress incident**, **Decompress emergent** options は **Edit DIFFRACT parameters** dialog box へ送られる。

17. ディスクの編集（Disk Editor）

17.1 序論

名前から分かるように、DiskEditor は、DIFFRACT™ の DISK 命令に用いられているように、種々のタイプの光ディスクの設定と変更を可能とする。

次の図は Writable ディスクを含む DiskEditor のメインウィンドウを示す。



新しいディスクは scratch から作られ、現在のデザインは保存されたり、外部ファイルまたは Windows のクリップボードから読み込まれる。最新のオプションでは DiffractGUI からディスクの構造を DiskEditor にコピーしたり、変更したディスクを DiffractGUI に書き戻すこともできる。

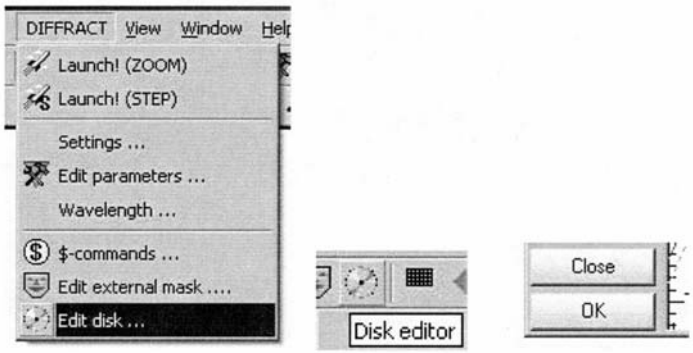
編集中のディスクの種々の構造とパラメーターは図面上で、または希望する値をタイプすることによって変更することができる。結果のディスク表面は直ちにグラフィックウィンドウに表示される。両方の方法は本章で後に述べる。
現在では、DiskEditor は DIFFRACT™ Ver.8.4 マニュアルに述べているディスクのみをサポートする。詳細はそのマニュアルを参照されたい。

グラフィックディスプレイはマウスのスクローリングホイールでズーミングでき、マウスの左ボタンでドラッギングしてパンできる。ドラッギングとは、マウスの左ボタンを押してマウスカーソルを動かすことを意味する。

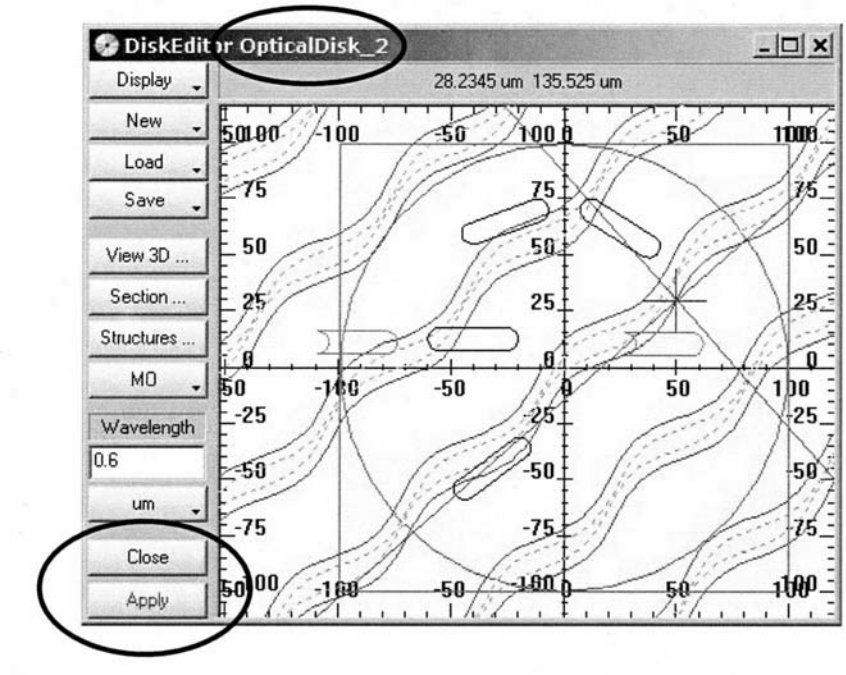
17.2　ディスクエディターの起動（Launching the DiskEditor）

DiskEditor は 3 つの方法で起動できる。(1) メニューから、(2) Controle Bar の DiskEditor アイコンをクリックする、(3) もし Edit Diffract Parameters dialog box が Disk コマンドを持っているときには EDIT Diffract Parameter dialog box の EDIT ボタンを押す。

方法 1 & 2：メニューから、あるいは Control Bar Icon から起動。



方法 1 & 2: 前に選んだ DISK に付属する Control Bar Icon から起動する。

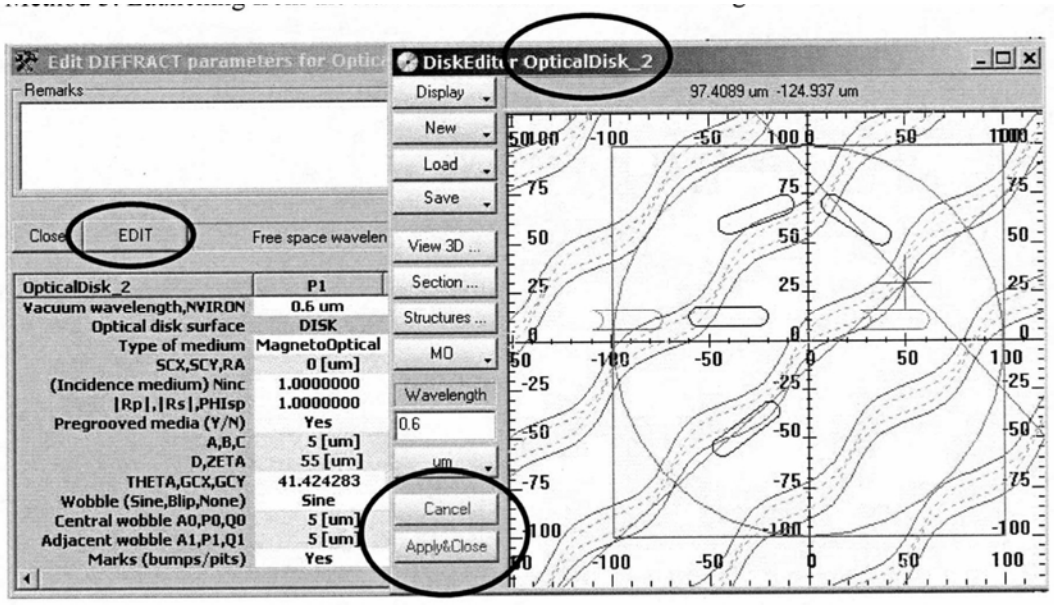


DISK が Workbench で選択されたとき、DiskEditor が方法 1 あるいは 2 で起動されたとすると、DiskEditor はその選択された DISK で初期化される。もし DISK が選択されないときは、DiskEditor は初期化されないで残り、DiskEditor は disk script file を作ったり、読込んだり、編集したり、保存する独立の module として使われる。この場合は、左側の一番下の 2 つのボタンは Close と OK と読み替える（上の図を参照のこと）。

もし前に選択された DISK コマンドのデータで DiskEditor が初期化されたとすると、左側の下にある 2 つのボタンは Close と Apply と読み替える。さらに、DISK の名前が DiskEditor の caption bar に現れ、DISK コマンドブロックが DiskEditor に残る。Apply はそのデータを Workbench の DISK に戻す。Close はデータ伝達しないで（たとえば最後の Apply の実行以後のすべての変更を廃棄して）Editor を閉じる。

方法 1 と 2 は DiskEditor を modeless window として起動する。それは DiskEditor が開いているときは、他の DiffractGUI のコマンドを実行できることを意味する。DiskEditor の実行中に付加された DISK が Workbench から除去されたときは、DiskEditor は開いたままであるが、Apply ボタンは何の作用もしない。さらに、1 つ以上の DiskEditor の事例が 1 つのそして同じ DISK について実行できる（そして開いたままにする）。この場合は Apply をクリックする度に、まえの DISK データを上書きする。

方法 3：Edit DIFFRACT パラメーターdialog box から起動する。

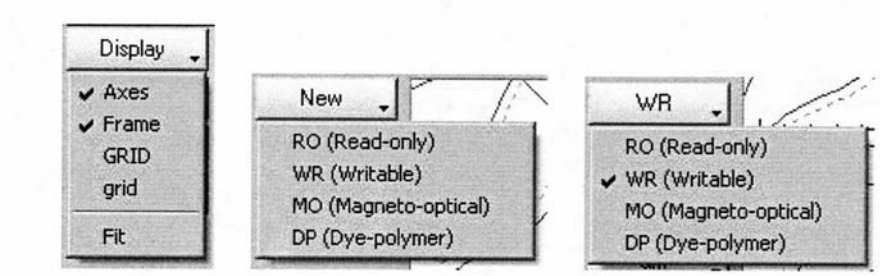


このように起動した場合、ただ 1 つの違いは DiskEditor はモードウインドウとして開かれることである。これは、作業を続けるために DiskEditor は Cancel または Apply&Close ボタンのどれかによって閉じられることである（ウインドウの右上にある closing x ボタンは Cancel と同じ効果を持つ）。

DiskEditor を起動する方法に関わらず、この章に述べたすべての control element は同じように働く。

17.3 The Main DiskEditor Window

DiskEditor の main window には次のような control element がある。



Display－はグラフィック出力に X-Y 軸、ラベルをつけたフレーム、および 2 種類のグリッドを加えることができる。Fit オプションはすべてのディスク面の構造*をカバーするためにグラフィックディスプレイに x-min, x-max, y-min, y-max を調整する。ディスク自身のパラメーターはこれらのオプションで影響を受けることは無い。

New－は決められたタイプ（RO, WR, MO, あるいは DP）の新しいディスクを作り、その表面に必要な構造をおく。これは今後の処理を助ける。
Caution: 新しいディスクを挿入する前に、このコマンドは何の警告も無く DiskEditor の内容を消去する：したがってすべての以前の変更は失われる。

RO, WR, MO あるいは **DP**: このボタンは編集中のタイプを表示する、さらに重要なことは、ディスクタイプの変更を受け付ける。しかし、ディスクの構造は演算の途中で消去されるので、これらの変更は充分慎重に行うべきである。たとえば、もし WR ディスクが溝を持っているとするとこれは RO ディスクに転換され、RO ディスクは溝構造をサポートしていないので、溝は自動的に削除される。この削除された構造は永久的に失われ、すなわち、上述の例に戻って、RO ディスクがらもとの WR ディスクに戻したとしても、削除された溝は自動的に回復されないことを留意されたい。

Units:－このボタンは現在使われている長さの単位を表示し、それを任意に変更することができる。もちろん、この操作は長さを表すパラメーターの数値を変更するが、他の数値は変更しない。さらに、グラフィックディスプレイは新しい値を反映してラベルを変更する。

* ディスク面の溝、凹凸のピットなどの記録に必要な構造を指す。

波長の値も同じ単位で表示される。

註：波長を長さの単位に選んだ場合、wavelength box はその結果常に 1 と表示される。

波長は対応する edit box に新しい値をタイプし、enter キーを押すか、OK ボタンを押して決定することができる。

Save と Load－は現在のディスクをファイルに記録するか、あるいはクリップボードに送る、あるいはディスクを外部ファイルあるいはクリップボードから導入することができる。一般に、DiskEditor は 2 つのフォーマットを取り扱う。両者は ASCII テキストファイルであって、DIFFRACT script および Disk script である。

DIFFRACT script フォーマットは最も有用であって、下の text box に示す。

```
$_____________________________________________________________________ #
                                                                       #
  Vacuum wavelength (um): 0.600000000000          NVIRON:  1.0000000 #
_______________________________________________________________________#
    Optical disk surface: DISK                  (Length_Units: um) #
          Type of medium: WR                                       #
             SCX,SCY,RA:          0          0         100        #
 (Incidence medium) Ninc:          1                               #
  Pregrooved media (Y/N): Y                                       #
                 A,B,C:          5    35.7341     39.9772        #
               D,ZETA:      63.7126         20                    #
Reflection coeffs. RL,RG:          1        0.5                    #
          THETA,GCX,GCY:         45    3.77929    -3.02343        #
 Wobble (Sine,Blip,None): Sine                                    #
 Central wobble A0,P0,Q0:          5         80     143.652        #
Adjacent wobble A1,P1,Q1:          5         80     31.3761        #
      Marks (bumps/pits): Y                                       #
         Number of marks: 2                                       #
   Mark # 01:   MCX,MCY:    -43.6399   -19.9368                    #
                 THETA:         45                                #
          Round or Flat: R                                        #
  Length, Width, Height:         40         10          5        #
                 Ratio:        0.5                               #
 Reflection coefficient:          1                               #
   Mark # 02:   MCX,MCY:     8.14879    32.1663                    #
                 THETA:         45                                #
          Round or Flat: R                                        #
  Length, Width, Height:         40         10          5        #
                 Ratio:        0.5                               #
 Reflection coefficient:          1                               #
Inspect surface pattern: Y                                        #
             Xmin,Xmax:       -100        100                    #
             Ymin,Ymax:       -100        100                    #
    Color or Gray-scale: C                                        #
   Save data file (Y/N): Y                                        #
        File identifier: 1                                        #
   Propagation distance:         10                               #
     Propagation regime: DFRNL                                    #
                  Smax:          0                               #
```

このフォーマットで export したディスクは直接 command.dat に挿入できる。あるいは反対に command.dat ファイルからコピーして DiskEditor で読み込むことができる。

この理由から、DiskEditor のウインドウは常にこの形式でクリップボードに取り出すことができる。

このフォーマットは、DIFFRACT™ V8.4 の仕様が厳重に守られているので、さらに詳しい記述と例については DIFFRACT™ のマニュアルを参照されたい。

第 2 のサポートされるフォーマットは DISK script であって、DiskEditor でのみ使われる。これはディスクを記録するのに使われる。それは将来のプログラムの改定に耐える堅牢さのためである。そのようなファイルは必須のキーワード disk parameter script で始まり、key-line generating_name=structure_name で区切られたテキストのブロックが続く。ここで、structure_name はディスク表面に適用する構造のタイプを表す。構造の個々のパラメーターは対応するテキストブロックで papameter_name=value units の形式で決めることができる（順序はどうでも良い)。このとき単位は何でも良く、長さのパラメーターは例のように記入する。しかし、non-length parameters をセットすると、無視される。

次のテキストボックスは Disk script 形式を説明する。

```
disk parameter script
wavelength=0.600000000000 um
nviron=1.0000000
length_units=um
disk_type=WR

generating_name=Aperture
RA=100 um
SCX=0 um
SCY=0 um
Ninc=1
propagation_distance=10 um
propagation_regime=DFRNL
Smax=0

generating_name=WRWobbles
A=5 um
B=35.7341 um
C=39.9772 um
D=63.7126 um
ZETA=20 um
THETA=45
GCX=3.77929 um
GCY=-3.02343 um
A0=5 um
P0=80 um
Q0=143.652
A1=5 um
P1=80 um
Q1=31.3761
RL=1
RG=0.5
```

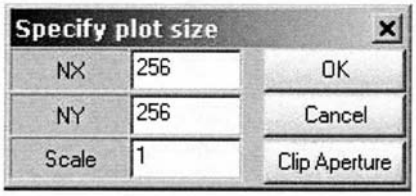
```
generating_name=WRMark
MCX=-43.6399 um
MCY=-19.9368 um
theta=45
round_or_flat=round
length=40 um
width=10 um
height=5 um
ratio=0.5
reflection_coeff=1

generating_name=WRMark
MCX=8.14879 um
MCY=32.1663 um
theta=45
round_or_flat=round
length=40 um
width=10 um
height=5 um
ratio=0.5
reflection_coeff=1

generating_name=Plot
Xmin=-100 um
Xmax=100 um
Ymin=-100 um
Ymax=100 um
save_to_file?=yes
color_or_gray?=color
file_id=1
```

DiskEditor がディスクをクリップボードから導入した場合、disk parameter script キーワードをチェックする。もし存在するときは Disk script フォーマットと考え、そうでなけれ

ば DIFFRACT™ script スタイルと決めると思われる。したがってどちらも導入することができる。しかし、DiskEditor のメインウインドウからクリップボードへ書き出したときはそのディスクデータは DIFFRACT™ フォーマットとなる。
ディスク表面を図面として保存

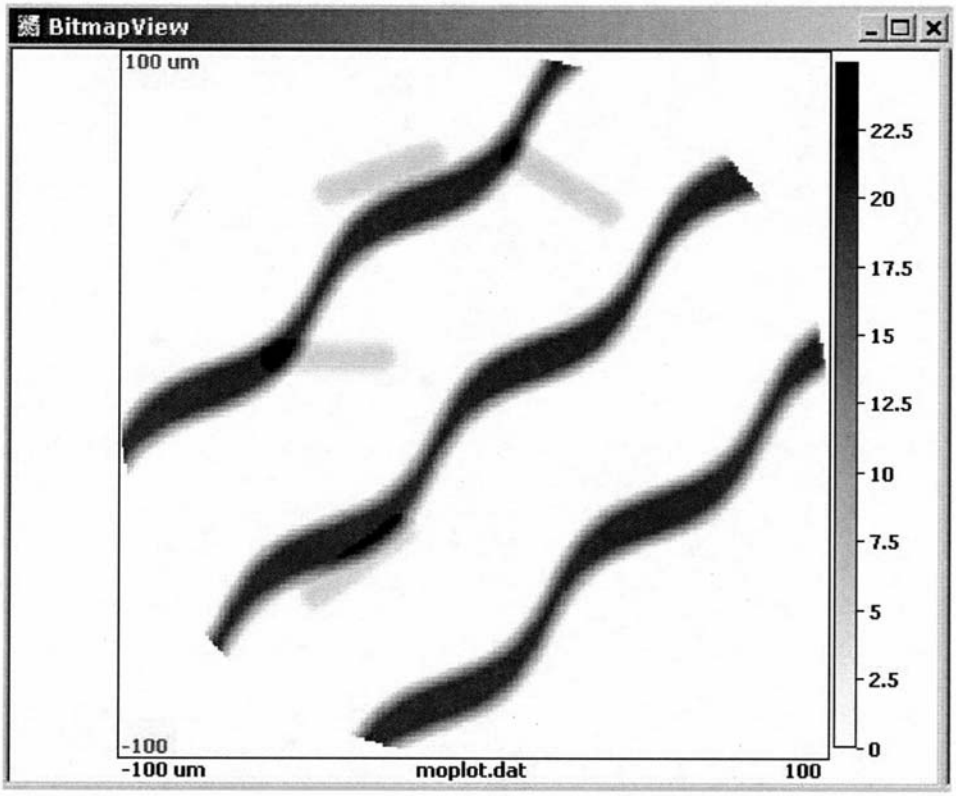
Save－Plot コマンドを用いて、表示された背景マップを DIFFRACT™ によって読み出された PLOT ファイルと同様に、数字のマトリックスとして読み出すことができる。これらのファイルは後で読み込むことができ、BitmapView あるいは DiffractGUI の ReliefView で見ることができる。

Save－Plot コマンドを起動した後、最初に小さい dialog box が開き、図のマトリックスの列（NX）と行（NY）を決めることができる。さらに、ディスク構造を切り取ってディスク開口の外へ取り出し、scaling factor を選んで図のファイルへ書き込む前に使うことができる。



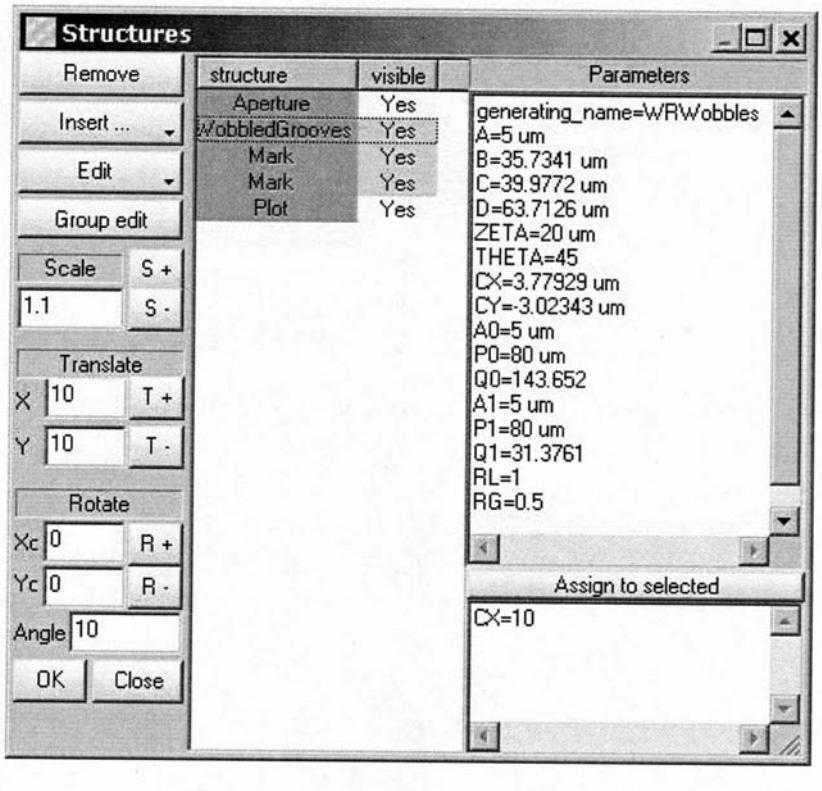
この特性は図と開口の構造を定義するとき必要であることに注意されたい。Clip Aperture オプションを選択しないときでも、開口は内部で必要である。下の図は DiffractGUI の BitmapView で観察できる結果を示す。

多層構造の DP ディスクの場合、それぞれの層の分離した図面ファイルを作ることができる。対応するファイル名は拡張子に"_layer_n"を加えることによって作ることができる。n は層の番号を表す。最後に、磁気ドメインはこれらの図面には含まれていないことを考えなければならない。それらは組み合わされたとき層の厚さに影響を及ぼさないからである。


BitmapView
100 um
22.5
20
17.5
15
12.5
10
7.5
5
2.5
0
-100
-100 um
moplot.dat
100

17.4 ディスク表面の構造（Disk Surface Structure）

個々のディスク構造は検査することができ、Structure dialog box で変更することができる。それは DiskEditor ウインドウの Structures…ボタンをクリックすれば開くことができる。



この dialog box は 3 つの部分から成り立つ。

- 左側には、controle elements があり、挿入、削除、拡大縮小、移動または回転が行われる。
- 中央には、structure-list があり、特定の構造を選択し、それぞれの構造を DiskEditor の画面で見るか、否かに従って、調節する。
- 右側には edit box があり、single structure（focused structure は下を見よ）のパラメーターのリストと、特定のパラメーターをすべての選択した structure に割り当てることができる。

structure-list 内では、セロ、1 個またはそれ以上の structure を選択することができる。しかしその structure の 1 つだけが特別な階層的な意味を持つ。この structure は次のように focused structure として引用される。リストの中では、focused structure は常に点線の矩形で囲まれ、parameter edit box (dialog box の右側にある)は、常にこの focused structure のパラメーターを表示する。

リストにある structure の 1 つを左ボタンクリックすれば、Ctrl および Shift キーの状態に関わらず、常にこの structure を focused とする。もしクリックの間 Ctrl も Shift も押されていなかったとすれば、その structure は選択されたとして mark され、他のすべての structures は unselected としてマークされる。もし Ctrl キーが押されていればクリックされた structure の選択状況は toggle される（すなわち、もしそれが選択されていれば、unselected となる。あるいはその逆の状態も起こりうる)。しかし、他のすべての structure には変化が無い。もし Shift キーが押されていれば、前の focused structure と新しく focus した structure との間のすべての structure は選択され、それ以外の structure は unselected となる。この選択機能は WindowsOS の FileExplorer で使われているのと同じである。また、focused structure を選択する必要も無い。

Visible コラムの Yes または No をクリックすると、グラフィックウインドウでその構造の visibility を toggle することになる。もし visible コラムヘッダーをクリックすると、すべての構造の visibility flag が toggle される。クリックの間に Curl キーを押し下げるとすべての構造は visible とマークされ、もし shift キーを押し下げると、すべての構造は not visible とマークされる。

最後に、グラフィックビューにディスプレイされている選択された構造は、その選択を示すために、太いペンで表示される。

ある構造の focused status が入力されると、DiskEditor はその構造のパラメーターリストを作り、それを（dialog window の右側にある）Parameter list box に表示する。focused status を閉じると、Parameter edit box の内容が構造パラメーターを実現するのに使われる。同じことが Stracture dialog box の OK ボタンをクリックしても実行される。

このことは、ユーザーによって行われた Parameter edit box の変更は OK ボタンか、異なった構造リストの構造へ focus すると、focused structure に反映される。単位(cm, mm, um, nm, あるいは波長）を考慮し、その値は常に DiskWindow で選択した単位に変換される。

Dialog window の左側にある controle element のうちで、最も重要なのは Remove と Insert ボタンである。Remove ボタンは focus された構造をディスクから削除する。しかし、これは selective manner で起こることがある。すなわち、すべての構造が削除されるわけではない。たとえば、Aperture はすべてのディスクタイプで必須であるため、これは削除してはいけない。さらに、DP ディスクの場合は、MarkSet 構造はマークの設定（すなわち凹凸のピット）の初期に必要であり、すべてのマークが前に削除されるまでは削除してはいけない。

新しい構造は Insert ボタンで挿入できる。これは次のようなオプションを提供する。

ディスクのタイプ、および既に存在している構造のタイプと数により、insert コマンドは現在の構成に対応するこれらの構造のオプションを可能とする。たとえば、もし WR ディスクが既に溝を持っていれば、第 2 の（サポートされていない）溝を挿入することはできない。

新しい構造は常に構造のリストの専用の位置に挿入される。典型的には、focus された構造は考えないが、これが必要な状況があり、そのときは考えなければいけない。たとえば、さらに 1 組のマークが必要な DP ディスクの場合、新しいマーク（凹または凸のピット）を focus されたマークの後で挿入される。この機能はどのようなマークを挿入すべきか調節できる。もし、このような状況において、focus された構造は Mark のタイプではないとすれば、新しい mark が現在存在する mark に付加される。もし mark のセットが定義されていないときは新しい mark も含み、mark の新しいセットが挿入される。

Scale, Translate および Rotate の controle element はすべての選択した構造に適用される。もし何の構造も選択しないときは、ディスクのすべての構造は影響を受ける。

Scaling については、Scale ラベルの下の edit box には 1 つのパラメーターが必要である。S+ボタンは選択された構造のすべての長さに掛け算している。一方、S－コマンドは長さを scaling parameter で割り算する（S+の逆）。

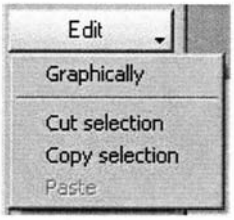
Translation では 2 つのパラメーターは必要である：X 方向と Y 方向である。それは X および Y のラベルのある edit box に設定する。Translation は T+ボタンあるいは T－ボタンで可能である。－は逆方向である。

Rotation では 3 つのパラメーターが必要である。すなわち回転の中心座標（Xc,Yc）および

回転角（度で表す）である。scaling と translation と同様に、R+は回転角、R－は R+と逆方向である。

一般に、これらの変換は物理的長さを表すパラメーターを変える。Translation と rotation パラメーターは実際は長さであるが、これらは長さの単位を変えたときにはいかなる変化も起こらない。したがって、この場合 T+,T－,R+,R－を使う前に注意が必要である。

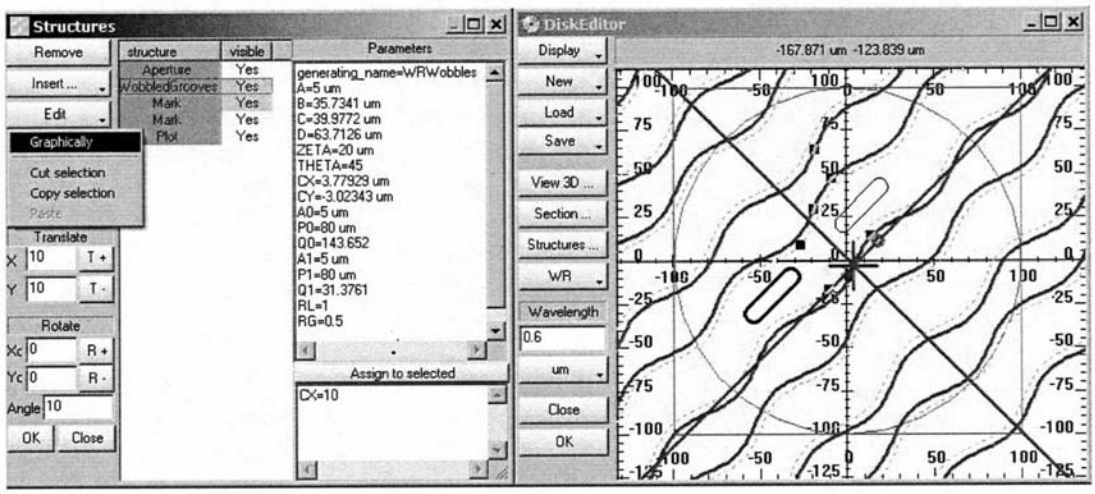
残った 2 つのボタン Group edit と Edit は構造を編集するとき有用であり、次に述べる。



Copy selection オプションは選択した構造をクリップボードにコピー（Disk Script 形式で、上を見よ）し、Cut selection も同様であるが、これはディスクから削除する。Paste はどのような文字もクリップボードから取り、Disk Script として解釈し、それを用いて存在する構造リストからディスク構造を組み立てる。この過程で、もし構造が現在のディスクタイプに許されないときは、それらは拒否され、挿入されない。

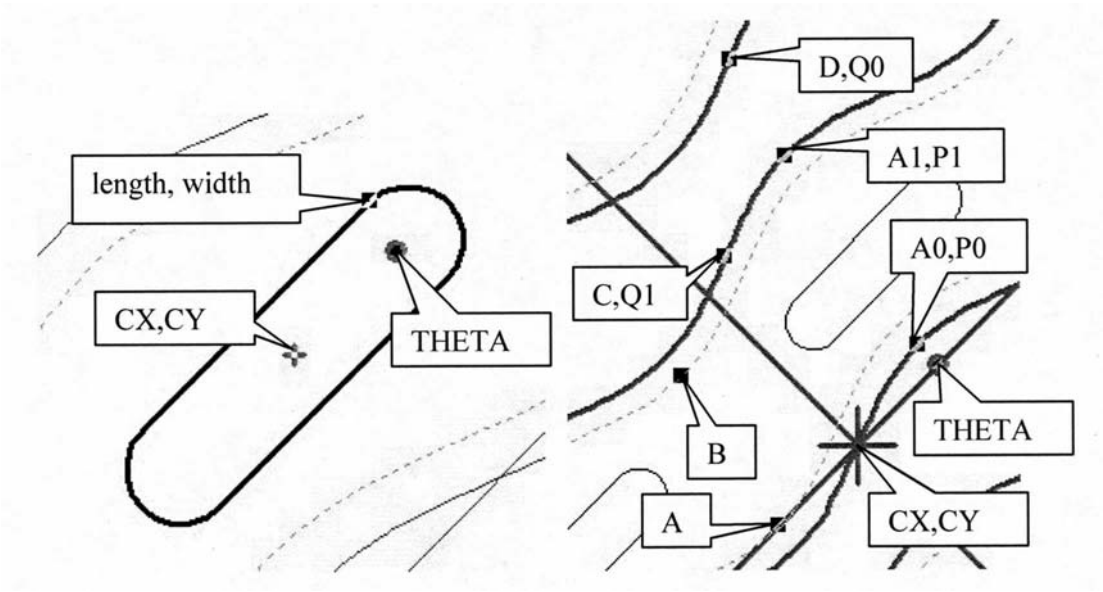
17.5　画像による構造の編集（Edit Structures Graphically）

この種のオペレーションは、下に示すように、DiskEditor ウインドウと同時に開いた Structure dialog box で最もよく実行できる。



グラフィックウインドウでディスプレイされた構造は特別な演算モード、すなわち edit-mode で Ctrl キーを押し下げ希望する構造の上でマウスの左ボタンクリックすることによって実行できる。もう 1 つの方法は、フォーカスした構造の edit-mode は Structure dialog box から Edit-Graphically ボタンを押すことによって入力することができる。この方法は溝とか、重なる構造に対して特に有効であり、このようにしてはグラフィックウインドウではアクセスできない。

次の図はマークとウオブル溝に対する edit-mode を示す。



このモードの明白な違いは、図の中の付加特性の番号（それらのほとんどは小さい矩形の中にある）が記入されていることである。これらの特性は edit node を参照されたい。これらの edit node は構造の 1 つあるいはそれ以上のパラメーターについて付加されている。

この node の形は、多少とも付加されたパラメーターのタイプを示す。

- 十字は構造の中心を示す。
- 円形は node が方向角に付けられている。
- 正方形はすべての他のパラメーターである。

それらの形に関わらず、すべての編集 node は同様に作用する。Node に付加されているパラメーターを変えるには、マウスカーソルをその node の上に置き、左ボタンを押しながら別の位置に動かす。マウスカーソルが動くと編集 node はカーソルに従い、その編集 node についているパラメーターは相応して変化する。マウスが動く間に変化は structure dialog box のパラメーターedit box に現れる。したがって、正確な数値が設定される。

編集モードから抜け出すためには、表示された編集 node 以外の部分をクリックすればよい、あるいは（上に述べたどれかの方法で）編集モードの他の構造に入ればよい。

同時に 1 つの構造のみが編集モードに入れることができる。しかし、構造ウインドウの Group edit ボタンをチェックすれば、1 つ以上の構造を同時に編集できる。この場合には、編集 node のドラッグによる変化は、すべての選択した構造に適用される。

たとえば、ディスク表面のマークのグループを移動させるときには、Group edit ボタンをチェックし、マークの 1 つ（どれでもよい）を編集モードに入れ、（CX, CY パラメーターについている）center node をマウスでドラッグする。すべての選択したマークは一斉に変化する。もちろん、もし θ の角、あるいは編集のマークの長さと幅が変化したときも同様である。

この特性は同じタイプ（典型的にはマークのグループ）の構造にのみ作動することを注意すべきである。

Plot と Aperture 構造は、マークの内部のどこでもクリックしてよいマークと対照的に、それらの描画の境界線をクリックする（また Ctrl キーを押し下げる）ことによって edit mode に入力することができる。Aperture と Plot はディスクの全アクティブ領域に広がっているためこの制限は必要であり、したがってすべての他の選択できない小さい構造をカバーする。

溝の中心点（CX, CY）は常にグラフィックウインドウのクロスとして表示される。溝は edit mode でこのクロスをクリックして（そしてもちろん Ctrl キーを押し下げて）表示すりことができる。

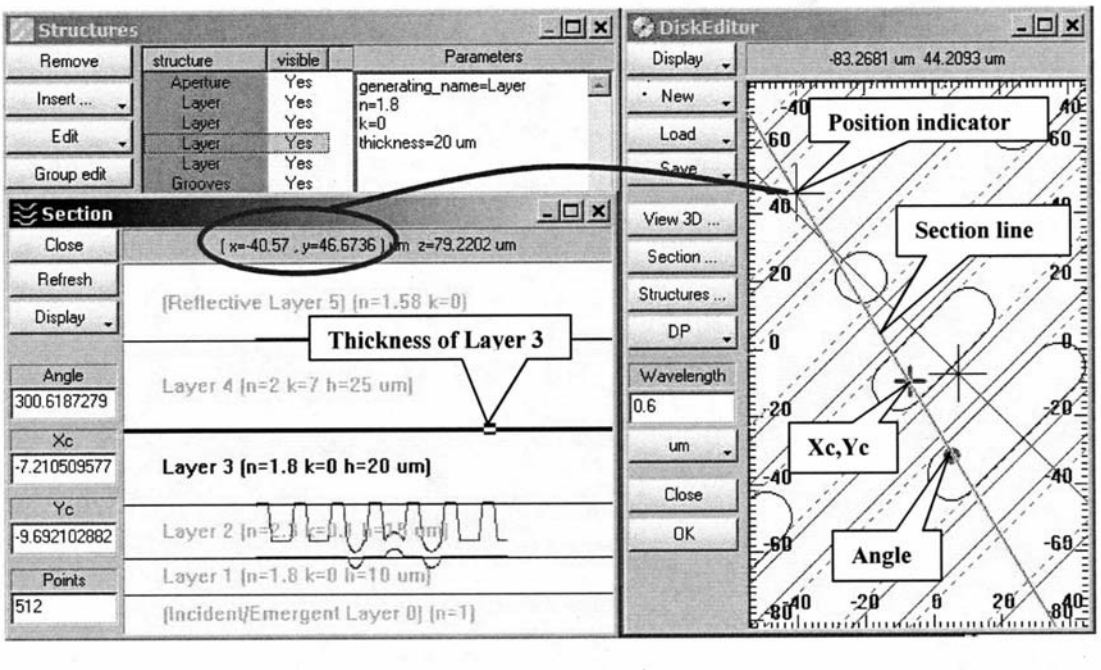
グラフィック編集による編集はディスプレイのピクセル構造により多少の粒状性が伴いことを考慮しなければならない。したがって、非常に精密な用途には Structure ウインドウのパラメーター編集ボックスを使わなければならない。

Assign to selected edit box は選択した構造のあるパラメーターと同じ値を決めることができる。これを行うためには、変更する構造を選択し、pamameter_name = value の形でパラメーターのリストを決め、Assign to selected をクリックする。この特性は、CX と CY パラメーターを同じ値に割り当てることによって、マークを水平に、または垂直に並び替えるのに役に立つ。

17.6　垂直断面（Vertical Cross Section）

DiskEditor ウインドウにある Section…オプションは構築中のディスクの任意の垂直 z 断面を見る機能を持っている。さらに、１層以上の場合はそれぞれの層の厚さを図形的に変化することができる。

Section…ボタンを押すと、2 つのことが起こる。Section dialog box が開かれ、DiskEditor グラフィックウインドウは黄色のラインを表示し、このラインに沿って z 断面が計算され、Section dialog box に表示できる。



断面の線はパラメーターXc, Yc,および Angle で決定される。これらのパラメーターは Reflesh ボタンをクリックして、あるいは図形的に他のディスク構造の編集と同様に Section dialog box で直接変更できる。同様に、切断線のための graphical editing に入るためには、Ctrl キーを押し下げ黄色の切断線で左マウスボタンをクリックすればよい。2 つの編集 node が現れ、それは上の図に示すように切断線のパラメーターを変えるために使われる。切断線のパラメーターが変るにつれ、Section dialog box の内容は連続的に更新される。切断線はディスクの表示領域内のみで計算されていることを注目すべきである。したがって表示の構造はこの機能を使う前に定義しなければならない。

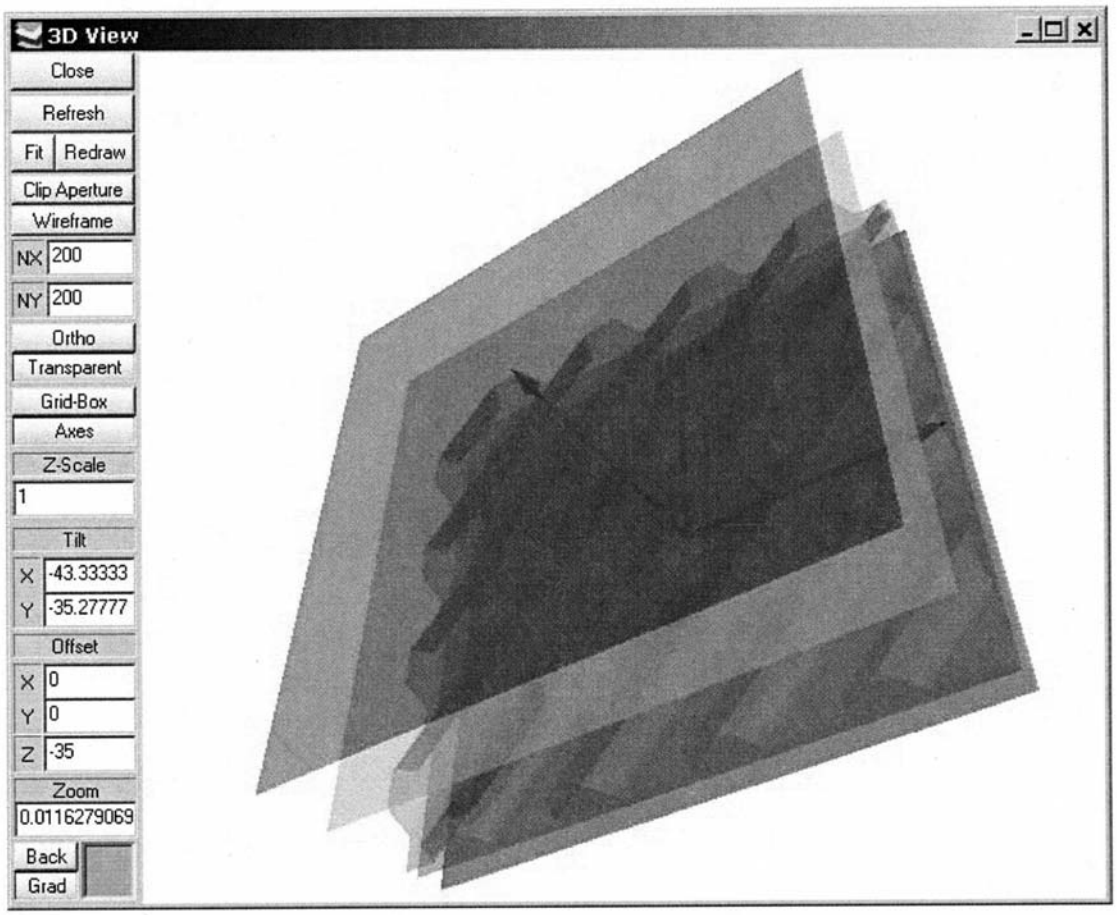
種々の層の切断面に加えて、Section dialog box が層の間の境界線も表示する。選択された層は太い文字と線で表示される。その厚さも上の図に表示したようにグラフィカルに編集できる。Structure dialog box の構造リストに visible flag を使って、各層は Section display

から削除できる。Point パラメーターは切断曲線の粒状性を決めることができる。

マウスカーソルは Section グラフィックの内部で動くので、Position indicator クロスは DiskEditor グラフィックウインドウ内に表示され、z 切断点の(x,y)位置を表示するため、切断線に沿って動く。

17.7　3 次元表示（3D View）

完全を期するため、DiskEditor ウインドウの View 3D…ボタンを使って、層の境界を 3 次元的に表示することができる。



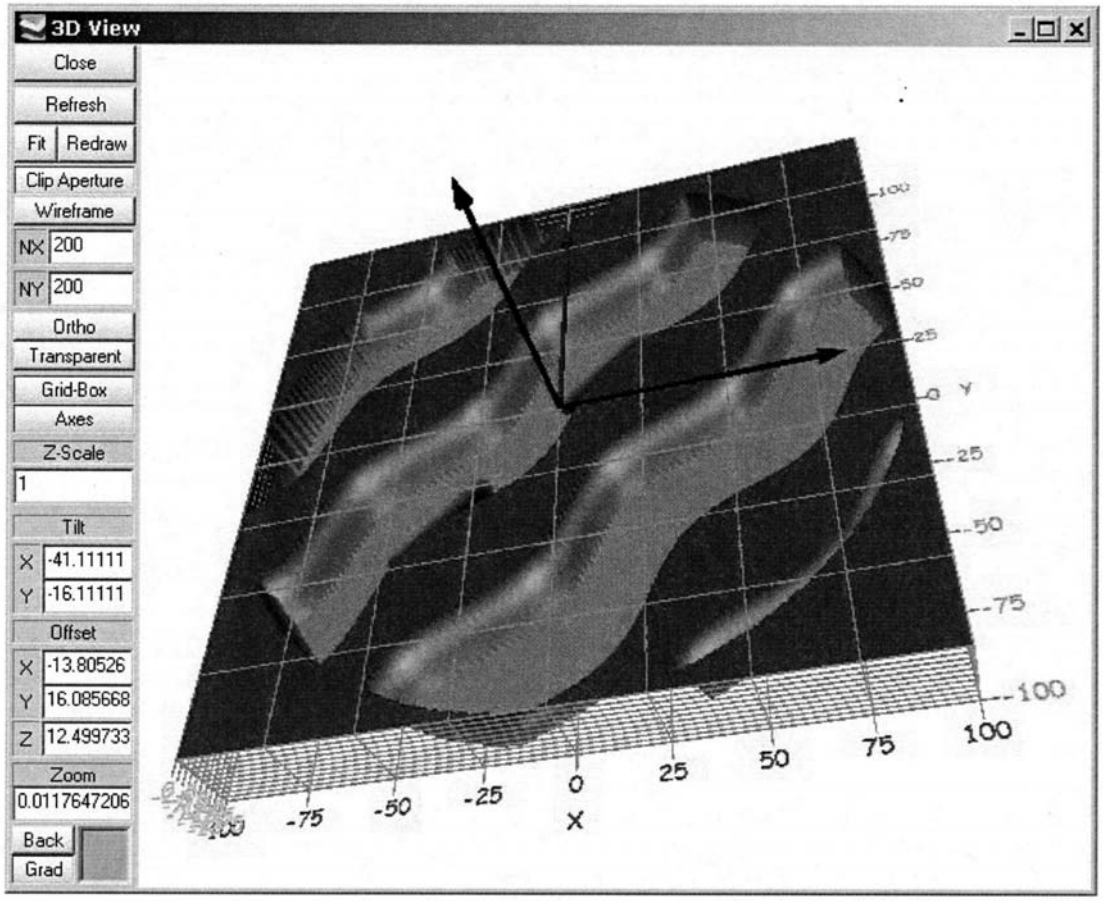
元来、ディスプレイは 3 つの座標軸を含んでいる。層の境界を表示するために、Reflesh をクリックすると、それぞれの層の内部データマトリックスを表示する。データを作り表示するのに比較的長い時間が必要なため、表示は DiskEditor, Structure あるいは Section ウインドウにおける変化に自動的に対応しない。したがって Reflesh ボタンをこれらの場合 3 次元表示を更新するのに使わなければならない。

データメッシュの粒状性は NX および NY で選ぶことができる。しかし、大きな値は法外な表示時間が必要なことを考慮しなければならないので、表示された表面の非常に小さい構造を使うようにしなければならない。

基本的に Redraw はスクリーンを修理できるが、必要ならば（たとえば NX または NY が変更されたとき）自動的に Reflech が実行される。

Z-Scale は z 座標のみに対する拡大率であり、小さい高さのよりよい visible structure を作ることができる。

Tilt, Offset および Zoom はディスクの方向を示すパラメーターであり、ディスプレイの面が基準となっている。*一般に、ディスクパラメーターのどれもがこの 3 次元表示では変更できない。*



Tilt の値は角度を表示し、3Dgraphics で左マウスボタンを使ってドラッグすれば変更できる。Offset 値は同様に変更できるが、Ctrl ボタンをドラッギング中は押し下げておかなければならない。2 つの offset 値のみがこの方法で変更できることを留意されたい。それらの

どれを変更するかは Tilt 角によって決まる。Zoom はマウスのスクロールホイールを回転すれば変更できる。もちろん、Tilt, Offset および Zoom は直接それぞれの edit box にタイプしても良いが、後で Redraw をクリックしなければならない。Fit はディスク全面が見えるようにディスプレイパラメーターを変更する。

Grid-Box は上に示すようにセットすることができ、座標軸の矢印は Axes チェックボタンで削除することができる。表面は Transparent ボタンで半透明にすることができ、rendering モードは Ortho ボタンで orthographic と perspective を切り替えることができる。背景の色は Windows default color に選ぶことができるが、Back ボタンでユーザー選択に変更できる。審美的見地から color gradient を Grad ボタンで適用することができる。画像は Clip aperture でディスク開口の中で表示でき、Wireframe ボタンで wireframe 表示とすることができる。

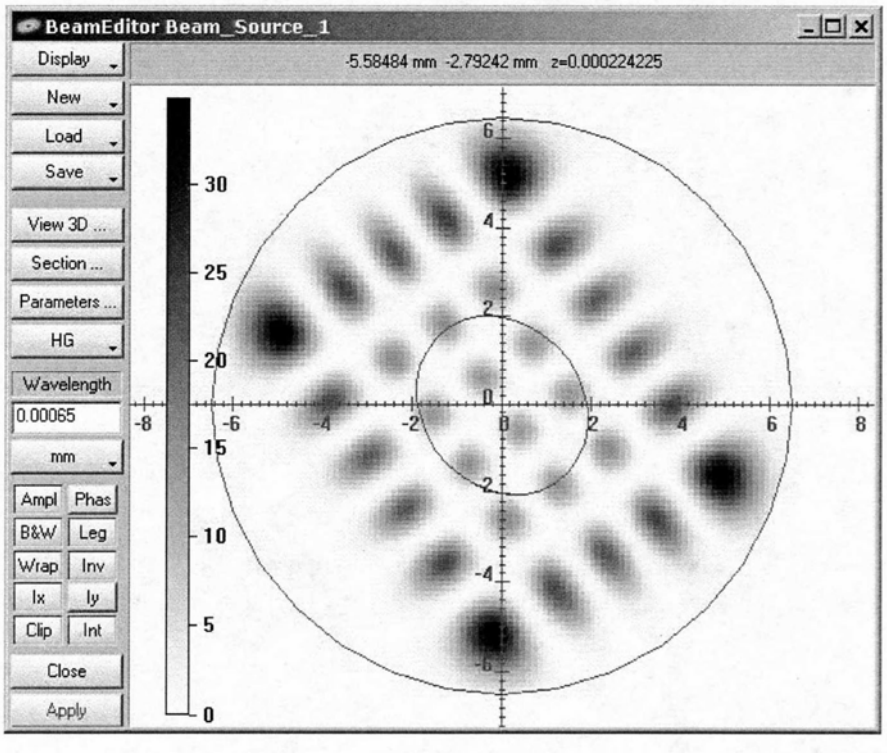
DiskEditor の他の特性については、Aperture を定義し、Plot で編集中のディスクの構造を表示する、そうでなければエラーメッセージが現れる。

個々の層の表面は、Structure dialog box の構造リストで visible flag No を選べば隠すことができる。

18　ビームエディター（Beam Editor）

18.1　序論（Introduction）

名前から分かるように、BeamEditor は、DIFFRACT™ BEAM コマンドと同じように、種々のタイプの BEAMS の設定と変更が可能である。次の図は HermiteGaussian ビームを表示した BeamEditor のウインドウを示す。



新しいビームはリストから選択し、そのデザインは保存したり、あるいは外部ファイルまたはクリップボードから導入することができる。最新のオプションでは BEAMs を DiffractGUI から BeamEditor にコピーでき、また編集した BEAM を DiffractGUI に書き戻すこともできる。

編集中のビームの種々のパラメーターは図形上または希望の値をタイプして編集できる。その結果のビームは直ちにグラフィックウインドウに表示される。両方の方法はこの章で後ほど記述する。

今のところ、BeamEditor は DIFFRACT™version8.4 マニュアルに述べたビームだけをサポートしている。詳細はそのマニュアルを参照されたい。

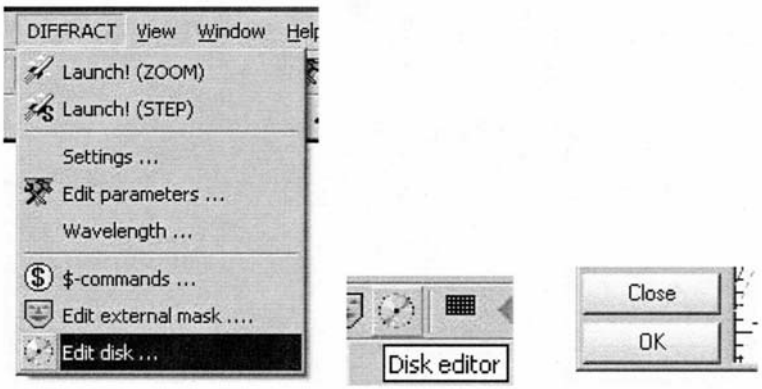
画像表示はマウスのスクロールホイールでズーミングでき、マウスの左ボタンでドラッグするとパンできる。ドラッギングはマウスの左ボタンを押しながらマウスカーソルを動か

せばよい。

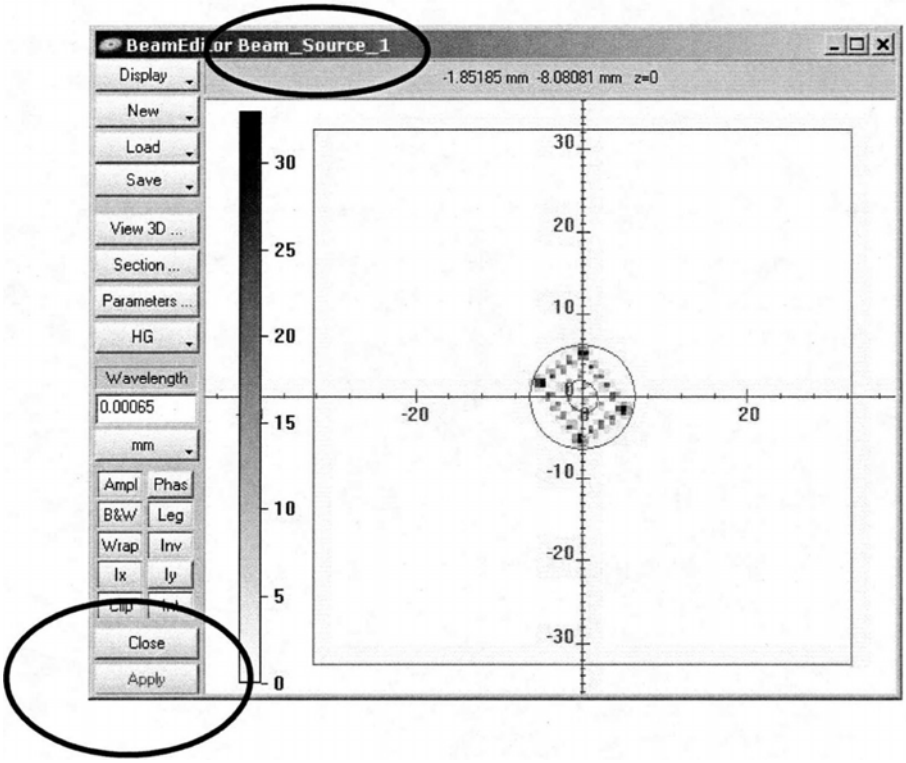
18.2　ビームエディターの起動（Launching the BeamEditor）

BeamEditor は 3 つの方法で起動できる。（1）メニューから、（2）ControleBar の BeamEditor アイコンをクリックする、あるいは（3）もし Edit DIFFRACT Parameters dialog box が BEAM コマンドを持っていれば、その dialog box の EDIT ボタンをクリックすればよい。

方法 1&2: Menu または ControlBar アイコンから起動。



方法 1&2:　前に選択した BEAM に付属する Control Bar Icon から起動。

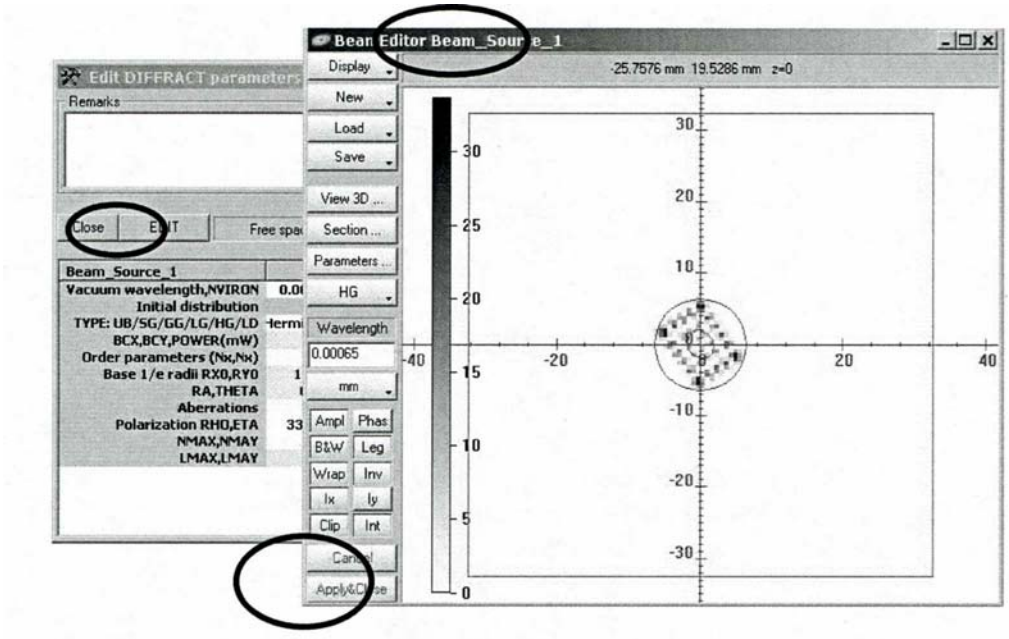


BeamEditor が特に選択されたビームで初期化されるのではなく、もし BeamEditor が方法 1 あるいは 2 で起動され、BEAM は workbench で選択されたとする。もし BEAM を選択しないでおくと、BeamEditor は初期化されず、BeamEditor は BEAM script ファイルを create/load/modufy/save する独立モジュールとして使われる。この場合、左側の一番下にある 2 つのボタンは Close と OK と読み替える（上図参照のこと)。

もし BeamEditor が前に選択した BEAM コマンドで初期化されたときは、左側の一番下のボタンは Close と Apply と読み替えていただきたい。さらに、BEAM の名前が BeamEditor の caption bar に現れ、BEAM コマンドブロックは BeanEditor に付属して残る。Apply はデータを Workbench の BEAM データへ書き戻す。Close はデータを移動することなく（たとえば Apply の最終的に実施された変化をすべて放棄して) Beam-Editor を終了する。

方法 1 および 2 は BeamEditor をモードなしウインドウとして起動する。これは Beam-Editor がオープンしているときは、他の DiffractGUI コマンドは実行できることを意味する。BeamEditor の実行中に Workbench から入力されている BEAM を削除すると、Beam - Editor は開いているが Apply ボタンはもはや何の作用もしない。さらに、1 つ以上の BeamEditor が 1 つで同じ BEAM に対してオープンする（オープンし続ける)。この場合、Apply のそれぞれのクリックは前の BEAM データを上書きする。

方法 3：Edit DIFFRACT parameter dialoh box から起動する。

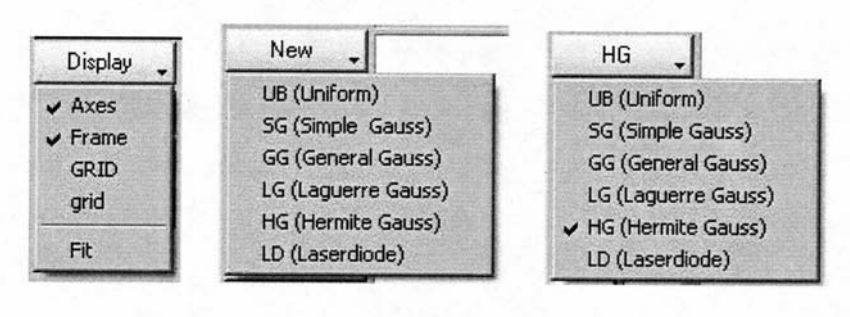


このような方法で起動したときの唯一の違いは BeamEditor は modal window としてオープンすることである。これは、作業を続けるために、BeamEditor は Cancel または

Apply&Close ボタンで閉じなければならない。（ウインドウの右上にある closing x ボタンは cancel と同じ効果を持つ。）BeamEditor を起動した方法によらず、この章に述べたすべてのコントロールは同じように働く。

18.3　BeamEditor のメインウインドウ（The Main BeamEditor Window）

BeamEditor のメインウィンドウは、次のコントロールエレメントを持つ。

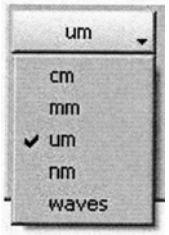


Display は X-Y 軸と、ラベルをつけたフレーム、および 2 種類のグリッドを図表の出力のために用意する。Fit オプションは、ビームメッシュをカバーするために、グラフィックディスプレイの x-min, x-max, y-min, y-max を調整する。BEAM のパラメーターそのものはこれらのオプションのどれによっても影響を受けない。

New は決められたタイプ（UB, SG, GG, LG, HG, あるいは LD）の新しいビームを作り、適当な初期値を設定する。

Caution は新しいビームを挿入する前に、BeamEditor の内容を予告なしに削除する。したがってすべてのこれまでの設定は失われる。

UB, SG, GG, LG, HG, あるいは **LD**：このボタンは編集中のビームを指示し、さらに重要なことは、BEAM タイプを変更することができる。しかし、これらの変更は注意しなければならない、BEAM パラメーターはこの作用によって消去されるからである。たとえば、もし UB (Uniform) BEAM がある種の収差を持っていれば、それは LD(Laserdiode)BEAM に変換され、LD-BEAM は収差を受け入れないので、収差の値は自動的に取り除かれる。削除された値は失われ復活できない。すなわち、上の例を考え、LD-BEAM が始の UB-BEAM に戻ったとしても、削除された収差の値は自動的に戻るわけではない。



Units:　このボタンは現在使っている長さの単位を示し、自由に変更できる。もちろん、この操作は長さを示すパラメーターの数値を変更し、他の変更は行わない。さらに、グラフィックディスプレイは新しい数値を反映してラベルを変える。波長の値も同じ単位で表示される。

註：長さの単位に wave を選ぶと、波長ボックスはいつも 1 の値を表示する。

波長は新しい値を対応する edit box にタイプし、Enter キーを押すか、OK ボタンをクリックすればよい。
BeamEditor の左下にチェックボタンのグループが表示されている。これらは背景のビットマップを作るのに利用できる。

Ampl および **Phas** は、もしビームの振幅または位相の分布が表示されたとき、区別する。もしこのチェックボタンのどれもチェックされないときは、背景ビットマップは表示されない。

B&W：はグレイとカラー表示を切り替えるのに用いられる。

Leg：はカラー凡例バーを出したり隠したりする。

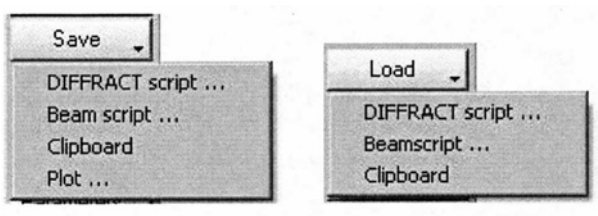
Wrap：このボタンは位相が表示され、収差が加えられたときのみ適用される。収差（Zernike または Seidel）は常に位相の値に転換される。もし、Wrap がチェックされているときは、これらの値は－180°と+180°に調整される。

Inv：チェックしたときはカラーlegend value は交換されている。これは、最大値は白（赤）でなく、黒（青）で表示される。

Ix, Iy：表示すべきコンポーネントを選択する。**Ampl, Phase** および **Int** ボタンの状態によって、表示された内容をよく反映するために、**Ix, Iy** のテキストラベルが **Ex,Ey** あるいは **Px, Py** に変化する。

Clip：もしこれをチェックすると決められた開口の外の表示された値がゼロにセットされる。

Int：振幅の分布が表示されている場合にのみ作動する。もし Int をチェックすると電気振幅は自乗され、そうでなければ作動しない。



Save と **Load**：は現在の BEAM をファイルに保存するか、クリップボードに送る。あるいは外部ファイルあるいはクリップボードから BEAM を読み込む。一般に BeamEditor は 2 つのフォーマットを取り扱う。両者は ASCII テキストファイルで DIFFRACT script と BEAM script である。

DIFFRACT script フォーマットは最も有用であって、下のテキストボックスに示す。

```
$_____________________________________________________________ #
                                                               #
  Vacuum wavelength (mm): 0.000650000000000   NVIRON:  1.0000000 #
_______________________________________________________________ #
    Initial distribution: BEAM                (Length_Units: mm) #
 TYPE: UB/SG/GG/LG/HG/LD: HG                                   #
       BCX,BCY,POWER(mW):          0          0          1     #
Order parameters (Nx,Ny):          4          5                #
  Base 1/e radii RX0,RY0:       1.82      2.145                #
              RA,THETA:          6.5         35                #
           Aberrations: None                                   #
   Polarization RHO,ETA:         33         22                #
              NMAX,NMAY:        256        256                #
              LMAX,LMAY:         65         65                #
```

このフォーマットで導入された BEAM は直接 command.dat ファイルに挿入でき、あるいは逆に command.dat ファイルからコピーして BeamEditor に読み込むこともできる。

この理由から、BeamEditor のウインドウは常にこのフォーマットでクリップボードに導出する。

このフォーマットで DIFFRACT™Ver.8.4 の仕様は厳密に尊重されすので、さらに詳しい記述や例については DIFFRACT™ マニュアルを参照されたい。

サポートされる第 2 のフォーマット、BEAM script は BeamEditor でのみ使われる。これは今後の Version 変更でも壊れにくいので、BEAM を記述・保存するために使うべきである。このファイルは必須のキーワード beam parameter script で始まり、generating_name= set_name というキーラインで導かれるテキストブロックで記述される。ここで、set_name は BEAM のパラメーターセットを意味する。セットの個々のパラメーターは対応するテキストブロック（順序は問わない）で parameter_name=value units の形式で設定する。ここに、units は任意で、長さのパラメーターに限り、次のように記述される。しかし、長さでは無いパラメーターを書くと無視される。

```
beam parameter script
length_units=mm
wavelength=0.65 um
beam_type=HG

generating_name=BeamMesh
NMAX=256
NMAY=256
LMAX=65 mm
LMAY=65 mm
```

```
generating_name=BeamGeometry_HG
BCX=0 mm
BCY=0 mm
Power_mW=1
Nx=4
Ny=5
RX0=1.82 mm
RY0=2.145 mm
RA=6.5 mm
THETA=35

generating_name=Polarization
RHO=33
ETA=22
```

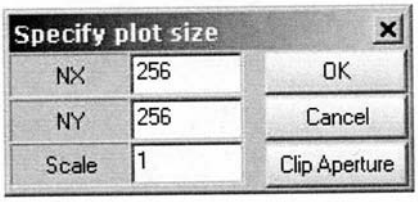
上のテキストボックスは BEAM script フォーマットを表示する。

BeamEditor がクリップボードから BEAM を導入すると、beam parameter sctipt キーワードをチェックし、もし見つかれば BEAM script フォーマットと判断し、そうでなければ DIFFRACT™ script スタイルと考える。したがって、どちらのフォーマットも導入することができる。しかし、BeamEditor ウインドウからクリックボードに送った場合、BEAM データは DIFFRACT™ フォーマットとして扱う。

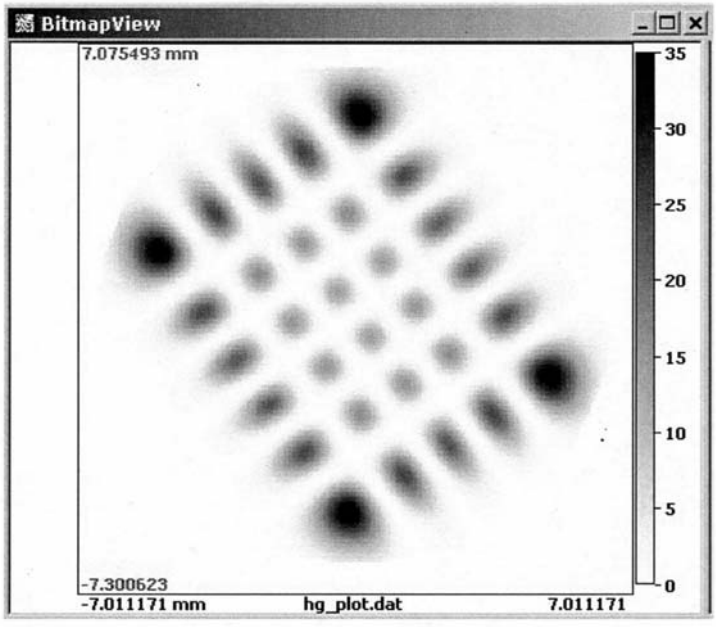
BEAM の分布を曲線として保存する。

Save－Plot コマンドを用いて、表示された背景マップは、DIFFRACT™ から導出された PLOT ファイルと同じように、数字のマトリクスとして導出される。これらのファイルは後で導入でき、BitmapView または DiffractGUI の ReliefView で表示できる。

Save-Plot コマンドを起動すると最初に小さな dialog box が開かれ、plot matrix の希望する列（NY）と行（NX）を決めることができる。さらにクリップするために BEAM 開口の外の BEAM 分布を選択し、scaling factor を選ぶことができる。これは plot file に書き込む前にデータに掛け算される。

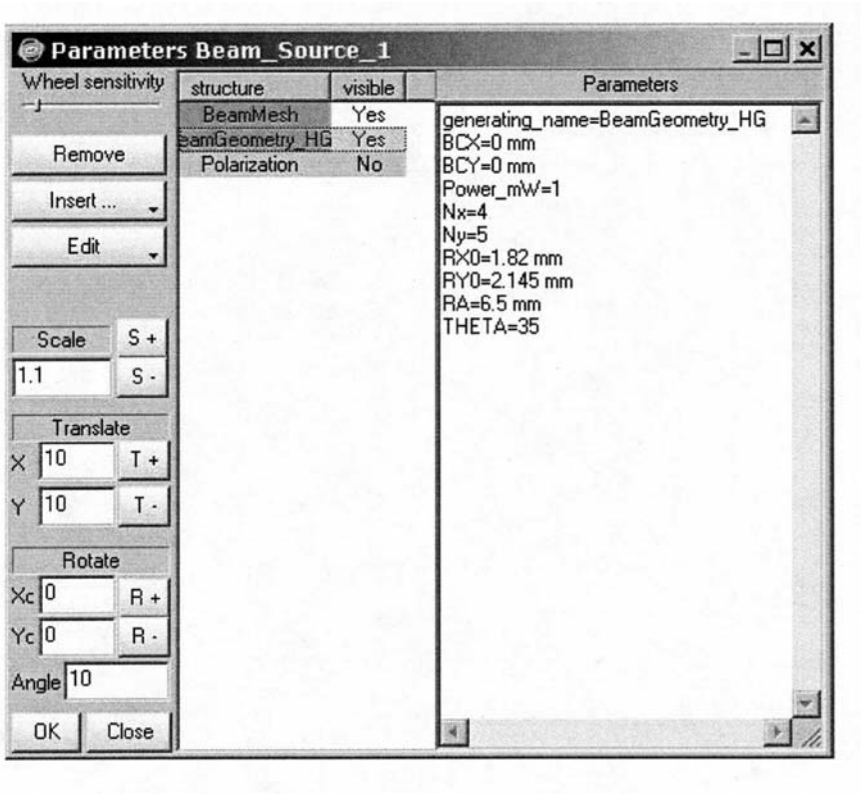


Plot は常に全表示面積に構築され、LMAX および LMAY によって与えられるメッシュ面内ではない。次の図は DiffractGUI の BitmapView で見られる結果である。



18.4 BEAM パラメーター（BEAM parameters）

個々のビームパラメターは検査され Parameters dialog box で修正できる。それは BeamEditor ウインドウの Parameters…ボタンをクリックすれば開かれる。



Dialog は 3 つの領域に分けられている。

- 左側にはパラメーターセットの挿入、削除、寸法の変更、開口やメッシュの正方形の移動、あるいは回転するコントロールエレメントがある。
- 中央にはパラメーターの特定のセットを選択したり、BeamEditor のウインドウにおけるグラフィック出力で見えたり、見えないようにするセットリストが表示される。
- 右側には 1 つのパラメーターセット（focused set、下を見よ）の編集ボックスを置く。

便宜上、BEAM のパラメーターは数個のセットにグループ化され、それぞれのセットは明瞭なパラメーターを持っている。次のようなセットが可能である。

BeamMech：は LMAX, LMAY, NMAX, NMAY を含む。すべての BEAM タイプに共通である。

BeamGeometry_XY：このセットはそれぞれの BEAM タイプで異なったパラメーターで作り、すべての対応する分布と開口を作るパラメーターを含んでいる。XY は UB, SG, GG, HG,

LG あるいは LD の記憶の 1 つである。

Polarization : は RHO（ρ）および ETA（η）の変更パラメーターを含む。すべての BEAM タイプに共通である。

Seidel および **Zernike** : これらのオプションセットは必要があればすべての BEAM タイプ（LD を除く）に可能であって、Seidel または Zernike 収差係数を包含している。

Set-list 内では、ゼロ、1 つ、あるいはそれ以上を選ぶことができる。しかしセットの 1 つは特別な階層的な意味がある。このセットは次のようにして **Focused** set とされている。リストの中では focused set は常に点線の矩形で囲まれていて、パラメーターedit box（dialog window の右側）は常にこの focused set のパラメーターを表示している。

リストにあるセットを左ボタンクリックすれば、Ctrl および Shift キーの状態に関わらず、常にそのセットを focused とする。クリックの間 Ctrl と shift キーの両者を押し下げた状態では、そのセットは選択されたとマークされ、他のセットは非選択となる。もし Ctrl キーを押し下げた状態では、クリックされたセットの選択状態は toggled となる（すなわち、もし選択された状態のときは非選択となり、その逆も成り立つ）が、他のすべてにセットの選択状態は変らない。もし **Shift** キーを押し下げると前に focus したセットと新しく focus したセットの間にあるすべてのセットは focus され、その間以外のセットは変化しない。この選択メカニズムは Windows OS の File Explorer に用いられているのと同じである。また、focused set をさらに選択する必要はないことに注意するべきである。

Visible コラムにおける Yes または No をクリックすると、グラフィックウインドウにおけるそのセットの visibility を toggle する。もし、**visible** コラムヘッダーをクリックするとすべてのセットの visibility フラッグは toggle される。もしクリックのとき Ctrl キーが押されているとすべてのセットは visible とマークされ、もし shift キーが押されているとすべてのセットは not visible とマークされる。

最後に、グラフィックビューにディスプレイされている選択されたセットは、それらの選択を反映するため、濃いペンで描かれる。

セットの focused-state を入力すると、BeamEditor はそのセットのパラメーターリストを作り、それを（dialog window の右側にある）Parameter edit box に表示する。

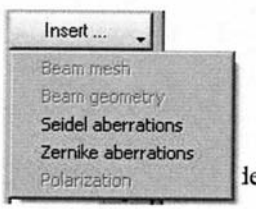
セットの focused-state をスイッチオフにすると、parameter edit box の内容は set

parameter を actualize するために使われる。もし parameter dialog box の **OK** ボタンをクリックしても同じことが起こる。

これは Parameter edit box においてユーザーによって作られた変更は、OK をクリックするか set list において異なったセットにフォーカスすると、focused set に戻される。単位（cm, mm, um, nm または波長）を考えると、値は常に BeamEditor ウインドウで決められた単位に変換される。

dialog window の左側にある controle element のうちで、最も重要なのは **Remove** と **Insert** である。Remove ボタンは BEAM から parameter の focused set を消去する。しかし、これは選択したものだけであって、すべてのセットが削除されるわけではない。たとえば、BeamGeometry パラメーターセットはすべての BEAM タイプに必須であり、それを削除することはできない。

新しいパラメーターは **Insert** ボタンで BEAM に挿入することができる。それは次のようなオプションを提供する。



*典型的にはこの特性は収差パラメーターセット（Seidel あるいは Zernike）を加えるのに最も有用である。その仕組みは両者を同時に加えるのを防いでいる。存在している収差のタイプを Seidel から Zernike に変える（その逆も）には、まず存在するセットを（**Remove** ボタンで）削除してから、新しいタイプを **Insert** ボタンで挿入する。*

Scale, Translate および Rotate はすべての選択したセットに適用される。もし何も選択していないときには、BEAM のすべてのセットに働く。

Scaling のためには”Scale”のラベルの下の edit box に 1 つのパラメーターを入れなければならない。**S+**ボタンは選択したセットのすべての長さに掛け算し、**S－**は長さを scaling parameter で割り算する（S+の効果の逆）。

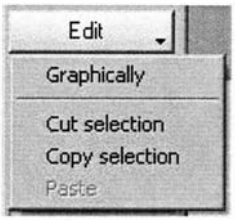
Translatiom は 2 つのパラメーターが必要である。これらは X および Y の移動ステップであって、これらは X および Y とラベルをつけた edit box で決めることができる。Translation step は **T+**または **T－**ボタンをクリックしても実行することができ、これらは逆方向に働く。

Rotation は 3 つのパラメーターを決めなければならない。回転の中心の座標(Xc, Yc)と回転角（度で表す）である。Scaling と Translation と同様に、**R+**は回転を実行し、**R－**は R+

と逆方向に回転させる。

一般に、これらの transformation は単に物理的長さを表すパラメーターを変える。Translation と Rotation パラメーターは事実長さであるが、それらは主とする長さの単位が変化したときは影響を受けない。したがってこの場合 **T+, T−, R+, R−**ボタンを使う前に注意しなければならない。

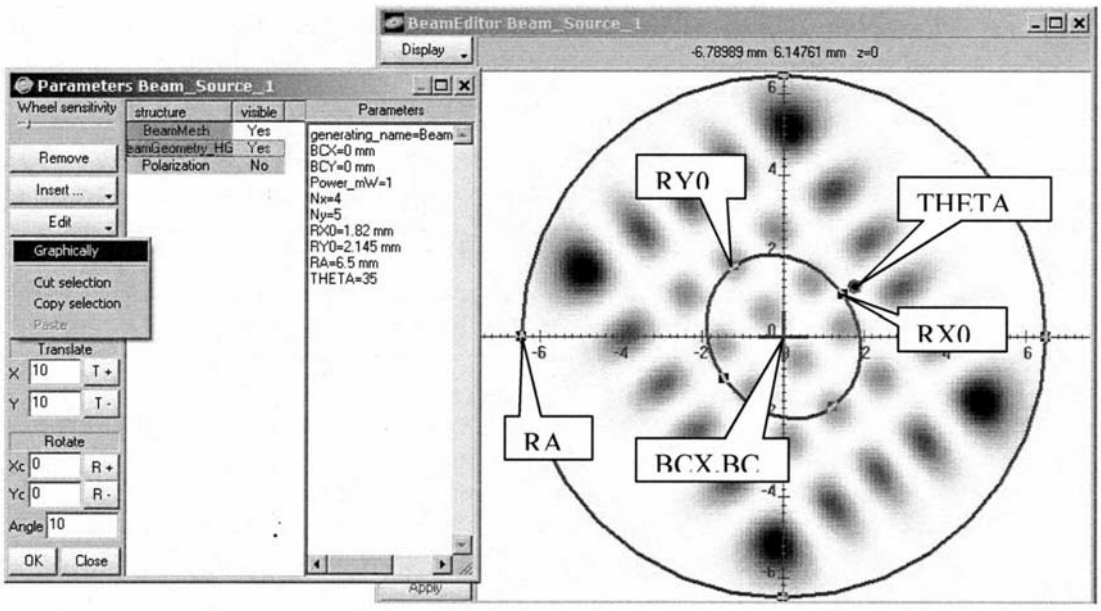
最後に残ったボタン **Edit** は構造を編集するとき有用であり、次に詳述する。



Copy selection オプションは選択したセットをクリップボードにコピー（BEAM script フォーマットで、上の記述を参照のこと）し、**Cut selection** は同じことをするが、BEAM からそれを削除する。**Paste** はクリップボードから文字を取り出し、BEAM Script として解釈し、それを既存のセットリストに挿入し、それによって BEAM セットを構成しようとする。この過程で、セットが現在の BEAM タイプに受け入れられなければ、放棄し、挿入されない。

18.5　パラメーターの図表的編集（Edit Parameters Graphically）

この種の処理は、下に示すように、BeamEditor ウインドウと同時に開かれている Parameter dialog box で実行するのが良い。



Graphic Window に表示されているそれぞれのセットは、Ctrl キーを押し下げ、希望する構造の上でマウスの左ボタンをクリックすると、特別な演算モード、edit mode, におくことができる。あるいは、focused structure の edit-mode は Parameter dialog box の Edit-Graphically ボタンを押しても良い。これは特に溝のような構造、または重なった構造に有用であり、graphic window からではこのような access はできない。

このモードの明らかな違いは、この図におけるいくつかの付加的特性（そのほとんどは小さな矩形である）の存在である。これらの特徴は edit nodes である。これらの edit nodes のそれぞれは、構造の 1 つまたはそれ以上のパラメーターに付けられている。
node の形は、ある程度、付加されたパラメーターおタイプを示している。

- 十字は形の中心を示す。
- 円は、node は方向角に付けられていることを示す。
- 矩形はその他のパラメーターに用いられる。

これらの形に関わらず、すべての edit node は同様に作用する。node に付けられたパラメーターを変更するために、マウスカーソルをその node 上に置き、左ボタンを押しながら違った位置に動かす。マウスカーソルが動くにつれて、edit node はカーソルに従い、その edit node に付けられたパラメーターが変化する。マウスの動きの途中で、変化は parameter dialog box に反映され、正確な数値を表すことができる。

Edit mode から出るためには、表示された edit mode の外のどこかをクリックするか、（上に述べた方法のどれかによって）edit mode の他のセットに入ればよい。

同時に edit mode には 1 つのセットのみをおくことができる。

マーク内部のどれかの点をクリックして入るマークと違って、BeamMesh セットは edit mode に図の周辺をクリックして（さらに Ctrl キー押しながら）入ることができる。

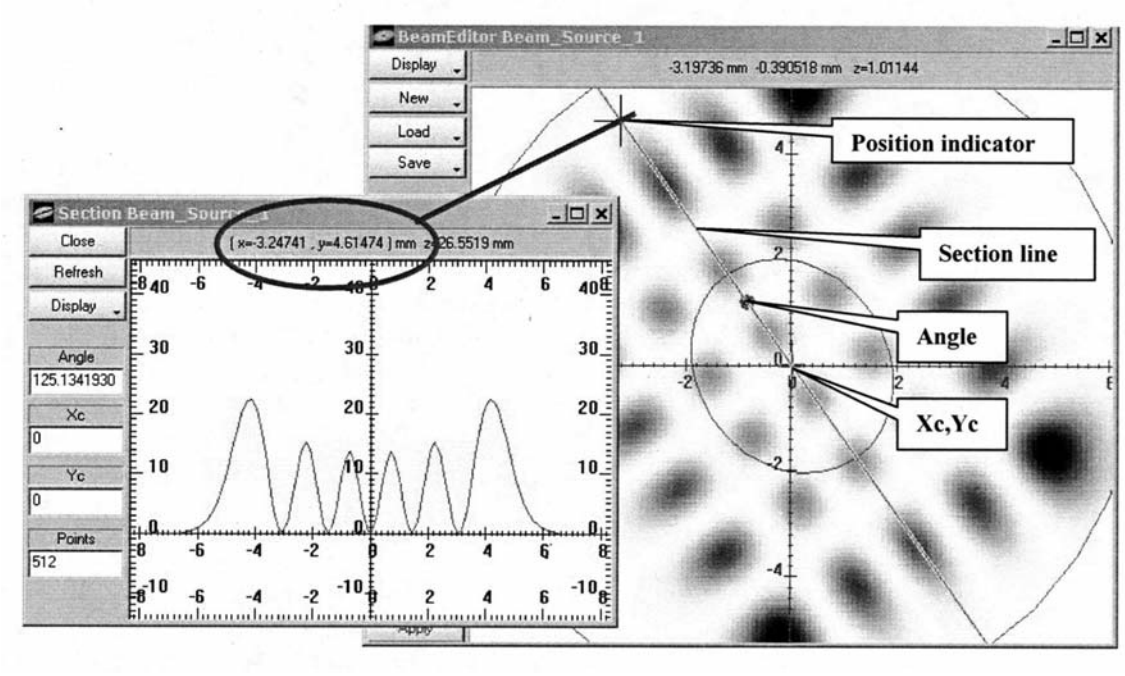
形の中心点はグラフィックウインドウの中に十字として表示される。対応するセットは edit mode にこの十字をクリックして（もちろん Ctrl キーを押しながら）入ることができる。

Graphic efitor による変更は、ディスプレイのピクセル構造のため、粒状性に何らかの変化を及ぼすことを留意すべきである。したがって、非常に正確な値に対して、Parameter ウインドウの Parameter edit box を使うべきである。

パラメーターを変更する第 3 の方法はマウスホイールを使うことである。この機能を使うときは、希望するパラメーターを含むテキスト行の中の Parameter edit box にテキスト訂正記号を入れ、Ctrl キー押したままホイールを回す。この状態でホイールの 1 クリックでパラメーターの値は数%だけ変化する。使用するパーセンテージはウインドウの左上部にあるホイール感度スライダーで 0.1～10%の間変化させることができる。この機能はゼロの値を変化させることはできないことに注意すべきである。この場合には、Ctrl キーと Shift キーを同時に押し下げると、ある決められたパーセンテージ（ホイール感度の設定によって 0.001～0.1 となる）を定数としてパラメーターに（掛け算の代わりに）加えることができる。

18.6 垂直断面（Vertical Cross Section）

BeamEditor のウインドウの Section…オプションは、構成中の BEAM 分布の垂直 z 断面を表示することができる。Section…ボタンをクリックすると、2 つの事柄が起こる : section dialog box が開かれ、BeamEditor グラフィックウインドウが z 断面を計算する黄色の直線を表示し、Section dialog box に表示する。



切断線は Xc, Yc、および角度のパラメーターで決めることができる。これらのパラメーターは、Refresh ボタンをクリックして、直接 Section dialog box で修正することができる。あるいは他の BEAM 構造の図表的編集と同様に図表的に編集できる。2 つの edit node が現れ、これらは上の図に示すように切断線のパラメーターを変更できる。切断線のパラメ

ーターが変化するに従って、Section dialog box の内容は連続的に update される。切断面は計算のための BEAM のメッシュ領域(BeamMesh set の LMAX と LMAY できめられる)で計算されることに留意すべきである。

Points パラメーターは切断曲線の粒状性を決める。

マウスカーソルが Section グラフィックの中を動くにつれ、Position indicator クロスが表示され、z 切断点の位置（x, y）を示すため BeamEditor グラフィックウインドウの中の切断線に沿って動く.

Section dialog box の表示はマウスのスクロールホイールで任意に拡大（zoom in）・縮小(zoom out）できる。さらに、もしホイールを回転しながら Ctrl キーを押し下げると、x 軸のみがズームされ、Shift キーを押し下げると y 軸だけがズームされ、このようにして任意のズーミング領域を確保できる。これは Main BeamEditor window においても有効であることを注意されたい。

18.7　3 次元表示（3D View）

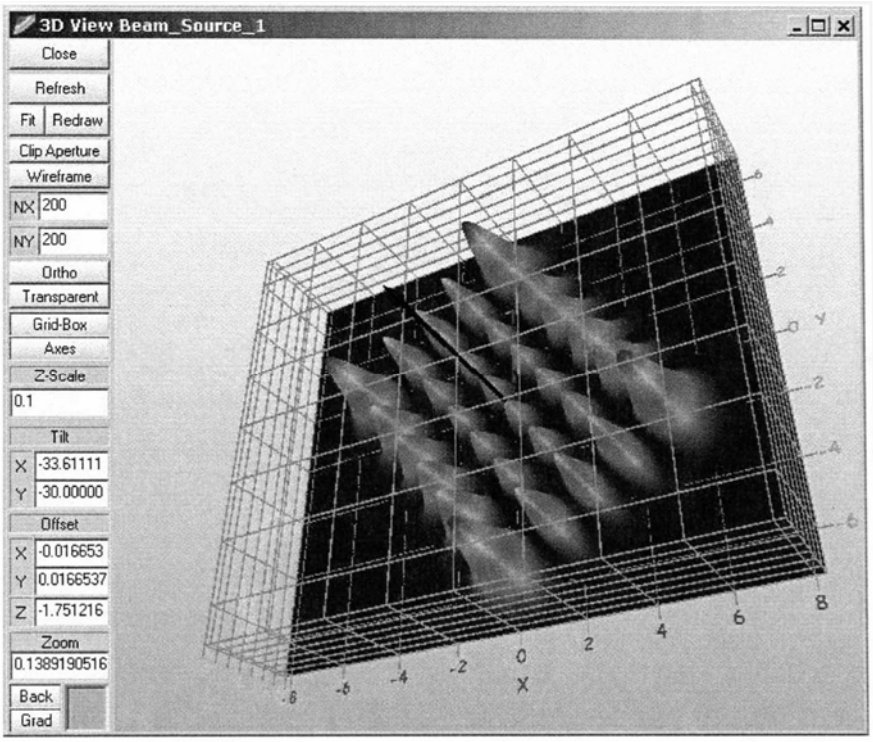
完全を期するため、BeamEditor ウインドウの View 3D…ボタンによって、分布は 3 次元的に見ることができる。元来、表示は 3 つの座標軸のみを持っている。分布を表示するためには Refresh をクリックすると、内部データマトリックスをそれぞれの層に対し表示する。データを作り表示するために比較的長い時間が掛かるため、表示は BeamEditor、Parameter, あるいは Section の変更による変化に自動的に追随できない。したがって、この場合 D 表示を更新するため Refresh ボタンを使うべきである。

データメッシュの粒状性は NX と NY で選ぶことができる。しかし、大きな値は法外に長い表示時間を要するので、表示表面で非常に小さい面積を見えるようにするとよい。

基本的に、Redraw はスクリーンを描き直すが、もし必要であれば、それは自動的に Refresh（たとえばもし NX と NY が変更されたとき）を実行する。

Z-Scale は z 座標のみを拡大する係数であって、高さを小さくして見やすい構造とすることができる。

Tilt, Offset および Zoom は BEAM の方向を決めるパラメーターであり、単に表示だけのためである。一般に BEAM のどのパラメーターも 3D 表示では変更できない。



Tilt の値は角度を意味し、3D 表示で左マウスボタンをドラッグして簡単に変更できる。Offset の値も同様にして変更できるが、Ctrl キーをドラッグの間押し続けなければならない。2 つの Offsei 値のみが、この方法で変更できることに留意されたい。それらのうちどれを変更するかは、Tilt 角に依存する。ズームはマウスのスクロールホイールを回転すればよい。もちろん、Tilt, Offset, Zoom の値はそれぞれの対応する edit box に直接タイプすればよいが、その後 Redraw をクリックしなければならない。

Fit は表示パラメーターを変更し、それによって完全な BEAM を見ることができる。

Grid-Box は、上図に示すように、取り付けることができ、座標軸の矢印は Axes チェックボタンによって外すことができる。表面は Transparent ボタンで半透明にでき、Ortho ボタンで表示を orthographic と perspective とを切り替えることができる。背景の色は Windows default color またはユーザー定義の色と Back ボタンで変更できる。見栄えをよくするために、Grad ボタンで color gradient を適用することができる。Clip aperture ボタンで画像を BEAM 開口の中で表示でき、Wireframe ボタンで表面をワイアーフレームで表示できる。

19. ワークフロー編集（Workflow Editor）

DIFFRACT™ に装備されている$-command line は種々のループとか、他のコントロール

構造を作るのに非常に役に立つ。しかし、大きな command.dat ファイルとかもっと複雑なプログラムでは DIFFRACT™ 命令の変化のとき up to date を行うのが難しいことがある。この困難を無くし$-command-line のすべての可能性を引き出すために、DiffractGUI は Workflow Editor 機能を提供している。Workflow Editor では controle structure を高いレベルの C 言語に似た言語で小さいプログラムを作ることができ、それは内蔵されたミニコンパイラーによって機械的に command.dat ファイル内の$-command-line に翻訳される。Workflow Editor は workflow script language と呼ばれる言語の開発環境を備えこの言語で書かれたプログラムを特定の command.dat のプログラムに翻訳するクロスーコンパイラーを持っている。

このツールによって、それぞれの$-command-line を個々に DIFFRACT™ command のパラメーターに変換する必要は無く、workflow-script コンパイラーはこれを実行する。必要なすべては wordflow-script プログラムを書くことであり、Wordflow Editor にこのことを実行させ、command.dat の中に正しく挿入できる$-command-line に翻訳することである。典型的には、与えられたシミュレーションセットアップに対して DIFFRACT™ を起動する前に行うべき最終ステップとすべきである。

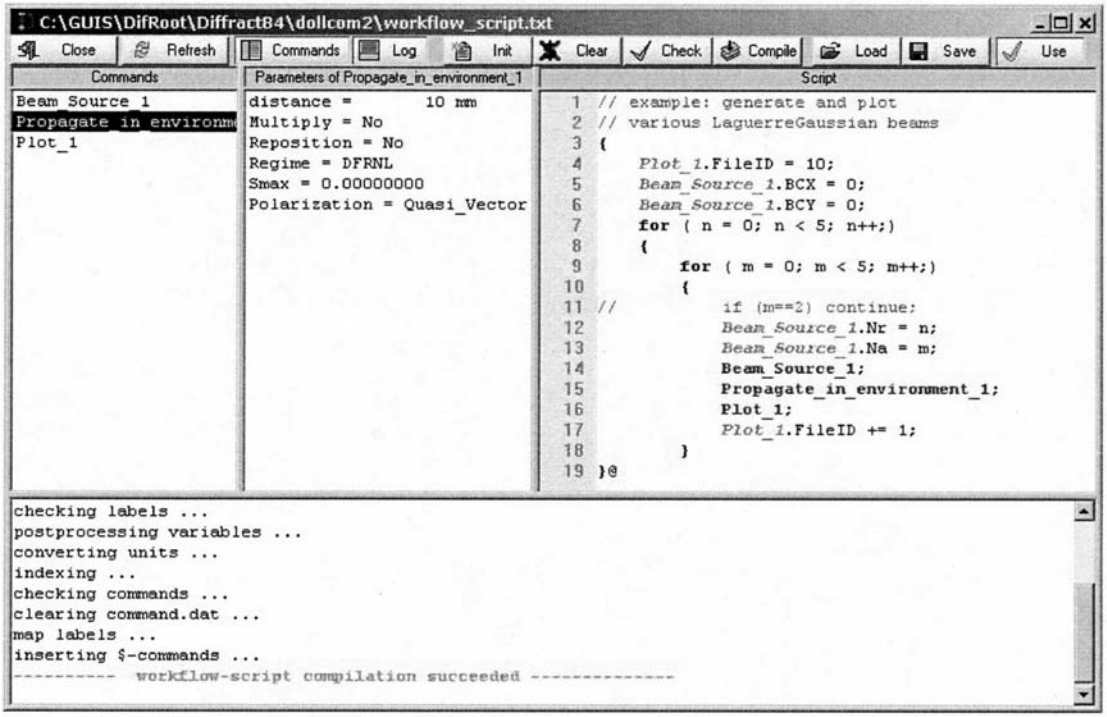
この章では Workflow Editor、workflow scripting language を説明し、簡単な例を挙げたい。

Workfloe Editor は **DIFFRACT－Workflow editor** メニュー、あるいは図に示すようにコマンドバーにあるショートカットアイコンをクリックして起動できる。

次の図はすぐ使える Workflow Editor を示す。4 つのウインドウを含む。

- 左側には、現在の command.dat ファイルにあるすべての DIFFRACT のコマンドを含む command list。このリストの中から 1 つのコマンドをマウスで選択する。

- 中央には、選択したコマンドの parameter list。
- 右側には最も重要なウインドウである workflow script に対する小さな text editor。
- 底部には script のコンパイリングにおける種々のメッセージやエラーを表示する。



ウインドウの最上部にあるボタンは次のような意味を持つ。

Close－Workflow Editor を隠す。この操作によって Workflow Editor の内容は変化を受けないことに留意されたい。ウインドウの右上にある×ボタンをクリックしても、同じ効果がある。

Refresh－workbench の現在の command を command list に加える。Workbench の変化を行った場合リストの更新を行う。

Commands－このチェックボタンは command list とパラメーター非表示を切り替える。もしそれらが隠されているときは、Script ウインドウがウインドウの全域に拡張される。

Log－Command ボタンと似たように作用するが、Log ウインドウに働く。両方のボタンは大きな script を編集する場合 Script ウインドウに大きな面積が必要なとき使うべきである。

Init－Script ウインドウにテキストブロックを作り挿入するときに用いる。ブロックは

command list の内容によって作られる。

Clear－現在の command.dat からすべての$-command を削除する。

Check－は、Log ウィンドウに適当なメッセージを書いて、現在の workflow を解釈し、検査する。これは Compile と似ているが、command.dat はこの操作で変更することはできない。現在の script がエラーがないことを検証することができる。

Compile－は最も重要な特性である。これは外部テキストファイルから script を導入したり、現在の Scipt ウィンドウの内容を外部ファイルへ保存することができる。Workflow Editor は Script ウィンドウの内容を常に workflow_script.txt と名づけられた特別なファイルに自動的に保存し、後で開いたときに元の script に読み戻す。したがって、Load および Save ボタンを全く使う必要は無い。しかし、これらは中間の、あるいは異なった version の script を記録するのに有用である。

Use－このチェックボタンは wrkflow script の使用を可能にしたり、不可能にしたりする。一般に$-command editor と Workflow Editor を同時に使うことは良くない（あるいは不可能）ので、どちらを使うかを決めなければならない。要求に応じて、DIFFRACT™ を起動する前に、DiffractGUI は workflow script を command.dat にコンパイルするので、このようにしてすべての手書きの$-command-line を上書きすることができる。これを避けるために、DIFFRACT™ を起動する前の準備でこのようにしてこれを無視するように DiffractGUI に知らせるために、Use ボタンはチェックしてはならない。他の微妙な違いは長さの単位の取り扱いである。もし Use ボタンをチェックすると$-command-editor は長さの単位を考慮しないし変換もしない。なぜならこれは Workflow Editor の仕事だからである。もし Use ボタンをチェックしないと、Worlflow Editor が存在しないように、$-command-editor は長さの単位を再び変換する。

19.1 Script Editor(The Script Editor)

Script ウインドウは小さなテキストエディターであり、いくらか進歩した編集機能を持っている。テキストは自動的にフォーマットされ特別な意味を表示するために色を付けられる。

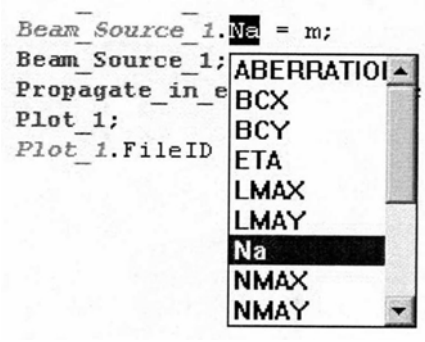
Green 文字は、comment area を表示する。**Blue Bold 文字**は workflow script language のキーワードを表示する。**Red Bold** 文字は DiffractGUI に読み込まれた command.dat にある有効な DIFFRACT™ コマンドの placeholder-name を指示する。最後に、**gray bold**

cursive（続け書き）文字はまた DIFFRACT™ コマンドであることを示すが、内部に enhannced variable name（次章を参照のこと）と結ばれていることを示す。混乱のために多くのスペースがないので、長さの単位は同じ gray bold cursive 文字で書かれている。

テキストブロックは通常の方法で選択できる。マウスカーソルで、または SHIFT キーを押してキーボードのカーソル矢印のどれかを押しても良い。テキスト全部は Ctrl+A で選択できる。1 行全部を選択するには、editor ウインドウの左にある行番号を 1 回クリックすればよい。選択されたテキストはマウスでドラッグ（選択したテキスト領域の中を左クリックし、マウスボタンを押し下げ、マウスを動かす）すれば簡単に動かすことができる。選択した部分を script 内の他の場所に（動かす代わりに）コピーするには、ドラッグのときに Ctrl キーを押し下げればよい。

Ctrl+C は選択したテキストをクリップボードへコピーし、Ctrl+V はどのようなテキストもクリップボードから現在のテキスト caret 位置に、もしあればそこで選択しているテキストと置き換えて、貼り付けることができる。

Workflow script language はいわゆる enhanced variables を、DIFFRACT™ コマンドに個々のパラメーターの specifier として受け入れる。それらは DIFFRACT™ コマンドの名前、次に”.”、最後にパラメーターの名前から構成されている。たとえば、”Beam_Source_1.**Na**”は”Beam_Source_1”コマンドのパラメーター”Na” を意味する。与えられたコマンドに対し、非常に多くの異なったパラメーターの名前があるので、script editor はここである種の援助を出す。Enhanced variable をタイプしているとき、”.”をタイプする度に、下に示すように、script editor は combo box を作り、可能なパラメーターのひとつを選択する便宜を提供している。



別の方法として、combo box はテキスト caret をドットの直前（DIFFRACT™ のコマンドとドットの間）においてキーボードの Ctrl+Space を押しても開くことができる。

Enhanced variable name を構築する第 2 のヘルプは **Commands** and **Parameters** リスト

ボックスから行うことができる。新しい enhanced variable を挿入するためには、その enhanced variable を挿入する位置にテキスト caret をおき、希望するコマンドを **command** list から選択する。**Parameters** list はそのコマンドのパラメーターで作られる。その **parameters** list に表示されたパラメーターの 1 つを左ダブルクリックすると、そのパラメーターの完全な enhanced variable name が作られ、自動的に script editor に挿入される。

最後に、**Workflow Editor** を閉じたときに、script editor の内容は常に working directory 内の workflow_script.txt という名前のファイルに記録される。このファイルは必要なときに新しく作られ、もし既に存在しているときには上書きされる。作業を始めるときには、script editor はもしあれば workflow_script.txt で満たされているが、そうでなければ空である。この機能性は違った working directory に違った script を使う可能性があり、現在の変更した script を保存することを考えなくても良い。しかし、どのファイルも手動で workflow_script.txt と名前をつけてはいけない。これは、その内容が次回の **Worlflow Editor** の終了のとき自動的に上書きされてしまうからである。

19.2 Workflow script の言語（The Workflow Scripting Language）

Workflow コンパイラーで認識される言語は C 言語の省略サブセットである。これは C 言語の広範な応用のために選択され、また C 言語の知識の無い読者にとっても学習・理解が容易と考えたからである。

装備された workflow scripting 言語は次のような reserved token-commands を持っている：

{, }, }@, for, else, break, goto, if, continue, do, while, tags, until

これに加えてコンパイラーは次の長さの単位を理解する：

mm, um, n,, cm, waves

さらに現在の script が応用された command.dat で定義された DIFFRACT™ コマンドの名前を受付ける。そのような DIFFRACT™ コマンドは stand-alone でタイプし、セミコロンが続く。DIFFRACT™ コマンドを他の表現の部分としてで使うことは許されないが、state-alone statements としてのみ許される。

最初に、いくつかの一般法則を述べる：

Script は(a = b+ l のような) statements と、(if (a<b) c=3;などの) control structures とから成る。もし、この文書で述べたのと違う場合は、それぞれの statements はセミコロン";"で終了しなければならない。Multiple statements は 1 行内におくことができるが、通常 script は各行に 1 乃至 2 の statements を置く方が読みやすい。さらに、必要があれば、statements は 1 行以上に拡張しても良い。

丸括弧 "{" は開始を意味し、"}" は statements のブロックの終了を表す。

Script は "{" で始まり、special token "}@" で終わる。これは"end of Program"である。この token はコンパイルを終了し、後に何が書いてあっても関係無い。Workflow Editor は最初の文字が "{" であり、最後の 2 文字が "}@" であるとチェックし、もしそうではないと内部的に "}@" を script をコンパイラーに送る前に付け加える。

19.2.1 変数と定数（Variables and Constants）

定数は 112, -0.123, 1.2e-5 などのような単純な数値である。その数値の後に長さの単位（cm, mm, um, nm, あるいは波長）が付く。したがってたとえば"100um"は "100cm" とは違う。長さの単位は定数にだけ用いられ、変数とか他の構造には使わない。

変数には ordinary と enhanced の 2 種類がある。Ordinary 変数は単純に alfanumerical 名（名前は文字で始まり、準備されているキーワードの 1 つではない）で決められる。例：I, j, k, Abc, var1, var2 など。

Enhanced variables は DIFFRACT™ コマンドパラメーターの placeholder である。それらは次のように決められている。

command_name, Parameter_name

パラメーター名とコマンド名の間の非常に重要な分離ドット"."に注意されたい。command_name は DIFFRACT™ コマンドの名前で無ければならないし、Parameter_name はコマンドのパラメーターを決めなければならない。この種の有効な変数を決めるために、さらに詳しいことは Script Editor の前の記述を参照されたい。

他の言語と違い、普通の変数は使う前に宣言する必要は無い。その代わり、script に初めて現れたとき割り当てられる。一般にそれらの値はゼロと初期化される。

すべての変数と定数は浮動小数点数として扱わなければないことに注意されたい。これは

この script にアクセプトされる唯一のデータタイプである。

表現、関数および記述（Expressions, Functions and Statements）

表現は１つまたはそれ以上の変数あるいは定数で構成され、必要に応じて丸括弧でまとめられ演算子で分離される。有効な演算子は代数的演算子+, －, *, あるいは/, Boolean operator &&(logical AND), ||(logical OR)、あるいは比較演算子＝(equal), <=, >=, <, >, あるいは != (not equal) である。

表現の内部では、**sin, cos, tan**, および **exp** 関数が用いられる。**tags** 関数を除いては、これらは Workflow Editor で受け入れられる関数である。**tags** 関数は 2 つの形式で現れる。

tags*(variable)* あるいは ***tags****(variable_1, variable_2)*

第 1 の形式は **tags**(variable, variable)と同じである。
この関数を表現の中で使うことはできない。それは stad-alone statement で使わなければならないが、たとえば

tags*(variable);* あるいは ***tags****(variable_1, variable_2);*

その唯一の目的は、script の中の command.dat において$TAGS コマンドを挿入するためのコンパイラーを決めることである。変数は ordinary かあるいは enhanced である。

記述(statements)はいくつかの形式があるが、大まかに言ってそれらは２つのグループに分割できる：assignments と controle structures である。Assignment statements はすべて一般的な形式

variable_name assignment_operator expression;

ここに variable_name は ordinary または enhanced variable、assignment_operator は=, +=, *=, －=, あるいは/＝の 1 つである。また expression は valid expression を表す。必須の終了のセミコロンに注意されたい。Assignement operator は何か補足の説明が要る。第１に operator=は明らかであるが、比較の＝operator と混同してはいけない。

相違を強調するために、変数 a が値 3 を持っていると仮定する。a = I とすると変数 a は値１を持つが、a = I を実行すると変数 a はなお値 3 ををを持っている。代わりに、表現 a = I

は false を評価する。

他の assignment operator は例によってよく説明できる。

a += expression;　は　a = a + expression;の代わりである。
a -= expression;　は　a = a - expression;　の代わりである。
a *= expression;　　は　a = a * expression;　の代わりである。
a /= expression;　　は　a = a / expression　　の代わりである。

可能なときはいつでも、短い置き換え形式を使うことを奨励する。これによってより短い$-command-line を作ることができるからである。

さらに 2 つの assignment operator、++ および— —である。Variable name を付加するときは増加(++)または減少（— —）で variable value を 1 だけ増加または減少する。たとえば、もし variable a が値 3 を持つとき、aa++;を実行すると、variable value は 4 となり、a— —を実行すると variable value は 2 となる。

19.2.3 Control Structures

If-else 条件実行は次のような方法で実施される。

if (expression) single_statememnt
if (expression) {if_block}
if (expression) {if_block} else {else_block}

もし、expression がゼロではない値だとすると、if_block(あるいは single_statement)が実行され、そうではないと、もしあれば、else_block が実行される。

Loop はいくつかの方法で作ることができる。

while (expression) { while_block }

まず、expression を評価する。もしそれがゼロでないと、while_block が実行され、expression は再評価される。この sequence は expression がゼロではない結果を示す限り続けられる。もし expression がゼロを示せば、プログラムは while-bloch に続く statement にジャンプする。

do { while_block } while (expression);

この while-loop はここでは同様に作動するが、while_block はどのような場合でも最小限 1 回実行される。

do { until_block } until (expression);

until_block は expression がゼロの値をとる限り実行される。換言すれば、until_block は expression がゼロとなるまで実行されるが、最小 1 回である。

最も複雑で万能的な statements は for-loops である ;

for (assignment_1 ; expression; assignment_2 ;) single_statement
for (assignment_1 ; expression; assignment_2 ;) { loop_block }

これらの関数性は次の pseudo-code で最もよく説明でき、理解される :

1. assignment_1 を実行する。
2. expression を評価する。
3. もし expression がゼロなら（これは false を意味する）、loop を終了し loop_block に続く次の statement を実行する。
4. もし expression がゼロではないと（これは true を意味する）、loop_block または single_statement を実行する。
5. assignment_2 を実行する。
6. step_2 に続く。

次の例は典型的な for-loop を示し、それは do_something block を i = 3, 4, 5 および 6 のように 4 回実行する。

for (I = 3; i<7; i = i + 1;) { …do_something … }

上に述べたどれかの loop 構造の内部で、break または continue statement は受け付けられる。break は loop を直ちに終了し、loop に続く statement のプログラムを続ける。これに対し、continue は現在の loop block を終了せずに、すぐ次の loop を実行する。

完全に行うためには、コンパイラーは単なる goto jump statemennt を認識する ;

goto lable_name;

ここに lable_name はアルファニューメリック文字で作られた jump-target を表す。変数名の場合には、最初の字は文字であり、ラベルはプログラム内で重なりが無く、reserved keywords の 1 つと一致してはいけない。ターゲットラベルは colon “:” （セミコロンではない）に続く名前として判別される。

使用例：

…code…
if (a<4) goto mylabel;
…more code …
mylabel:
… even more code …

Tutorial

次の短い tutorial は簡単な workflow scripts の作成・実行の参考となる。基本的な startup として、下に示すように、hermite-gaussian BEAM と 2 つの PLOT コマンド（1 つは強度分布、他の 1 つは位相分布）を含む command.dat を考える。

```
$_____________________________________________________________________ #
                                                                       #
  Vacuum wavelength (mm): 0.000650000000000            NVIRON:  1.0000000 #
_____________________________________________________________________ #
    Initial distribution: BEAM                          (Length_Units: mm) #
 TYPE: UB/SG/GG/LG/HG/LD: HG                                           #
       BCX,BCY,POWER(mW):  0.00000000  0.00000000  1.0000000           #
Order parameters (Nx,Nx): 4 5                                          #
  Base 1/e radii RX0,RY0:  1.8200000  2.1450000                        #
              RA,THETA:  6.5000000  35.000000                          #
            Aberrations: None                                          #
    Polarization RHO,ETA:  33.000000  22.000000                        #
             NMAX,NMAY: 256 256                                        #
             LMAX,LMAY:  65.000000  65.000000                          #
$                                                                      #
$_____________________________________________________________________ #
                                                                       #
  Vacuum wavelength (nm): 650.000000000                                #
      Logarithmic SCALE:  9.0000000                                    #
              Xmin,Xmax: -10000.000  10000.000                         #
              Ymin,Ymax: -10000.000  10000.000                         #
```

```
     Color or Gray-scale: C                                                  #
             Z-component: N                                                  #
   Save data files (Y/N): Y                                                  #
         File identifier: 0                                                  #
$                                                                            #
$____________________________________________________________________________#
                                                                             #
  Vacuum wavelength (nm): 650.000000000               NVIRON:  1.0000000     #
_____________________________________________________________________________#
       Plot distribution: PLOT                        (Length_Units: nm)     #
                    Type: Phase                                              #
       Logarithmic SCALE:  4.0000000                                         #
              Xmin,Xmax: -10000.000  10000.000                               #
              Ymin,Ymax: -10000.000  10000.000                               #
     Color or Gray-scale: C                                                  #
             Z-component: N                                                  #
   Save data files (Y/N): Y                                                  #
         File identifier: 1                                                 #
```

この例では、要求は BEAM の Nx = 0, 1, 3, 4 および Ny=0, 1 ,2, 4 の組み合わせによる図面を描く workflow script を開発するものである。Nx=2 と Ny=3 の組み合わせは省略している。さらに、それぞれの plot ファイルは Ixnm.dat および Pxnm.dat で、n=Nx, m=Ny である。

スタートするには Worlflow Editor を開き、Use ボタンがチェックされていることを確認し、Init ボタンをクリックする。これにより、script editor ウインドウは次のようなスケルトンーテキストが現れる。

```
  {
        Beam_Source_1;
        Plot_1;
        Plot_2;
  }@
```

Beam_Source_1, Plot_1, Plot_2 は command.dat（command list に表示されているように）と同じ順序で書かれている。この条件は重要であり、常に満たされなければならない。

Skelton-script を次のように変更する（行の始まり、間隔、およびコメントは任意である）；

```
{
for ( nx=0, nx<5; nx++; )   //   begin of the loop over nx
{
    if ( nx == 2 ) continue;   // skip nx = 2
    for ( ny = 0; ny<5; ny++; ) // begin of the loop over ny
    {
        if ( ny == 3 ) continue;   // skip ny = 3
```

```
            Beam_Source_1.Nx = nx; // assign nx to the Nx parameter of Beam_Source_1
            Beam_Source_1.Ny = ny; // assign ny to the Ny parameter of Beam_Source_1
            Beam_Source_1;
            Plot_1.FileID = 10*nx + ny; // assign proper FileID of Plot_1
            Plot_1;
            Plot_2.FileID = 10*nx + ny; // assign proper FileID of Plot_2
            Plot_2;
    } // end of the ny-loop
} // end of the nx-loop
}@
```

さらに、Beam_Surce_1, Plot_1, Plot_2 の順序が変わっていないことを注意することは非常に重要である。**Check** ボタンをクリックすると、script を確証することができる。もし syntax にエラーがないと、**Log** ウインドウは、他の情報メッセージの間に、

---------- workflow-script check succeeded ----------

のメッセージが表示される。そうではないと、エラーメッセージが表示され確証プロセスが停止する。

この時点で、command.dat はまだ変更されていない。command.dat に$-command-line を挿入するためには **Compile** ボタンをクリックされたい。Check と同じように、もしエラーがないと、Log ウインドウには次のメッセージが表示される。

--------- workflow-script compilation succeeded ----------.。

ここで、command.dat ファイルは使用できるので、それを保存し、DIFFRACT™ を起動すればよい。また、**Check**, **Compile** および保存ステップを省略し、script を終了した直後に DIFFRACT™ を起動する方法を選択してもよい。起動コマンドを実行する前に、もし **Use** ボタンをチェックすると、DiffractGUI は常に command.dat の変化または script の変化を見つけると workflow script をコンパイルすることを要求する。そのときは表示されたメッセージボックスの **Yes** ボタンをクリックして返事をしていただきたい。もしコンパイルのエラーが生じたときは、起動命令はキャンセルされ、適当なメッセージボックスが示される。

出来上がった$-command-line とラベルは$-command-editor または他の DiffractGUI の編集機能で調べたり、修正することができる。しかし、ほとんどの変化は予期しない結果を招来するので、workflow script compiler で作られた$-command-line あるいは line-label を変更または削除し**ない**ように強く要請する。

完全を帰するため、次の script が DIFFRACTTM を通る全部の command.dat のリストを作る。

```
$VALUE                    : 0                                                  #01
$VALUE                    : 1                                                  #03
$VALUE                    : 0                                                  #04
$VALUE                    : 0                                                  #05
$VALUE                    : 0                                                  #06
$MAKE#06,ITEM1=#04,ITEM1:                                                      #
$VALUE                    : 0                                                  #
$JUMP#12_IF#06,ITEM1.GE.: 5                                                    #
$JUMP#14_IF#06,ITEM1,NE.: 2                                                    #
$JUMP#13_IF#03,ITEM1,EQ.: 1                                                    #
$VALUE                    : 0                                                  #14
$MAKE#05,ITEM1=#04,ITEM1:                                                      #
$VALUE                    : 0                                                  #15
$JUMP#13_IF#05,ITEM1.GE.: 5                                                    #
$JUMP#18_IF#05,ITEM1.NE.: 3                                                    #
$JUMP#17_IF#03,ITEM1.EQ.: 1                                                    #
$VALUE                    : 0                                                  #18
$MAKE#09,ITEM1=#06,ITEM1:                                                      #
$MAKE#09,ITEM2=#05,ITEM1:                                                      #
$_____________________________________________________________________________ #
                                                                               #
  Vacuum wavelength (mm): 0.000650000000000            NVIRON:  1.0000000      #
_____________________________________________________________________________ #
    Initial distribution: BEAM                          (Length_Units: mm)     #
 TYPE: UB/SG/GG/LG/HG/LD: HG                                                   #
       BCX,BCY,POWER(mW):  0.00000000  0.00000000  1.0000000                   #
Order parameters (Nx,Nx): 1 1                                                  #
  Base 1/e radii RX0,RY0:  1.8200000  2.1450000                                #
                RA,THETA:  6.5000000  35.000000                                #
            Aberrations: None                                                  #
    Polarization RHO,ETA:  33.000000  22.000000                                #
              NMAX,NMAY: 256 256                                               #
              LMAX,LMAY:  65.000000  65.000000                                 #
$VALUE                    : 10                                                 #19
$MAKE#19,ITEM1=#04,ITEM1:                                                      #
$INCREMENT#19,ITEM1...BY: 10                                                   #
$MAKE#19,ITEM1*#06,ITEM1:                                                      #
$MAKE#19,ITEM1+#05,ITEM1:                                                      #
$MAKE#08,ITEM1=#19,ITEM1:                                                      #
$_____________________________________________________________________________ #
  Vacuum wavelength (mm): 650.000000000                NVIRON:  1.0000000      #
_____________________________________________________________________________ #
       Plot distribution: PLOT                          (Length_Units:   )     #
                    Type: Intensity                                            #
       Logarithmic SCALE:  9.0000000                                           #
              Xmin,Xmax: -10000.000  10000.000                                 #
              Ymin,Ymax: -10000.000  10000.000                                 #
     Color or Gray-scale: C                                                    #
            Z-component: N                                                     #
   Save data files (Y/N): Y                                                    #
```

```
        File identifier: 0                                                  #08
$VALUE                 : 10                                                 #20
$MAKE#20,ITEM1=#04,ITEM1:                                                   #
$INCREMENT#20,ITEM1...BY: 10                                                #
$MAKE#20,ITEM1*#06,ITEM1:                                                   #
$MAKE#20,ITEM1+#05,ITEM1:                                                   #
$MAKE#07,ITEM1=#20,ITEM1:                                                   #
$_____________________________________________________________________      #
                                                                            #
  Vacuum wavelength (nm): 650.000000000              NVIRON:  1.0000000     #
_____________________________________________________________________      #
       Plot distribution: PLOT                        (Length_Units:   )    #
                    Type: Phase                                             #
       Logarithmic SCALE:  4.0000000                                        #
              Xmin,Xmax: -10000.000  10000.000                              #
              Ymin,Ymax: -10000.000  10000.000                              #
     Color or Gray-scale: C                                                 #
            Z-component: N                                                  #
   Save data files (Y/N): Y                                                 #
        File identifier: 1                                                  #07
$_____________________________________________________________________      #
  Vacuum wavelength (nm): 650.0000000000             NVIRON:  1.0000000     #
_____________________________________________________________________      #
$VALUE                 : 0                                                  #17
$INCREMENT#05,ITEM1...BY: 1                                                 #
$JUMP#15_IF#03,ITEM1.EQ.: 1                                                 #
$VALUE                 : 0                                                  #13
$INCREMENT#06,ITEM1...BY: 1                                                 #
$JUMP#11_IF#03,ITEM1.EQ.: 1                                                 #
$VALUE                 : 0                                                  #12
         End of session: QUIT                                               #
@@@
```

```
// save a plot for each intermediary steps
        Beam_Source_1;
        SaveDistribution_1;
        Plot_1.FieldID = 20; // plot - Ids shall start at 20
for (z=5 mm; z<=1 cm; z += 1000 um; )
{ // begin of the  loop over z
        if (Plot_1.FieldID>99) break; // terminate loop if plot=ID exceeds limits
        Read_Distribution_1;
        Propagate_in_environment; _1.datanace = z; // assign distance
        Propagate_in_environment_1;
        Plot_1;
        Plot_1.FieldID++; // increment plot ID
} // end of loop over z
}@
```

この例は伝播距離を５ｍｍから 10ｍｍまで１mm 間隔で変化できる。最初の plot は FileID = 20、次は 21 など、最終距離 10ｍｍに達するまで実行できる。ここで、最終の plot は FileID=25 であるが、もし距離間隔を小さくすれば、最終の HileID は導大し、99（これは可能な ID の最大値）を越えることもできる。これを避けるため、break statement の例として、loop 内の if-statement は ID が 99 を越えると終了する。

script を通して、単位は 1 ではなく個々の DIFFRACT™ コマンドとして用いられる単位と同じものとなる。

また、非常に重要な留意点はこの script に用いられる DIFFRACT™ コマンドの order は、command.dat にリストされる order と同じである。これは workflow script の一般的な要求である。

for-loop statement を使うことは必要では無い。次の script は他の loop-constructs を使うことを説明する。これは前にリストした同じように働くが、異なった loop タイプを用いている。読みやすいように、loop のみを変えている。

while-loop 用いる script:

```
{
// example for metric units
// propagate a beam from z=5 mm to 10 mm in 1 mm steps
// save a plot for each intermediary step
    Beam_Source_1;
    SaveDistribution_1;
    plot_1.fieldID = 20; // plot Ids shall start at 20
    z = 5 mm; // assign start distance to z
while ( z <= 1 cm )
{ // begin of the loop over z
    if (Plot_1.FieldID>99) break; // terminate loop if plot ID exceed limits
    Read_Distribution_1;
    Propagate_in_environment_1.diatance = z; // assign distance
```

```
    Propagate_in_environment_1;
    Plot_1;
    Plot_1.FileID++; // increment plot ID
    z += 1000 um; // increment z by 1 mm
} // end of loop over z
}@
```

do-while-loop を用いる script:

```
{
// example for metric units
// propagate a beam from z=5 mm to 10 mm in 1 mm steps
// save a plot for each intermediary step
    Beam_Source_1;
    SaveDistribution_1;
    Plot_1.FileID = 20; // plot IDe shall start at 20
    z = 5 mm; // assign start distance to z
do
{ // begin of the loop over z
    if (Plot_1.FileID>99) break; // terminate loop if plot ID exceeds limits
    Read_Distribution_1;
    Propagate_in_environment_1.distance = z; // assign distance
    Propagate_in_environment_1;
    Plot_1;
    Plot_1.FileID++; // increment plot ID
    z += 1000 um; // increment z by 1 mm
} // end of loop over z
while ( z <= 1 cm )
}@
```

do-until-loop を使う script:

```
 {
// example for metric units
// propagate a beam from z=5 mm to 10 mm in 1 mm steps
// save a plot for each intermediary steps
    Beam_Source_1;
    SaveDistribution_1;
    Plot_1FileID = 20; // plot Ids shall start at 20
    z = 5 mm; // assign start distance to z
do
{ // begin of the loop over z
    if (Plot_1.FileID>99) break; // terminate loop if plot ID exceeds limits
    Read_Distribution_1;
    Propagate_in_environment_1.distance = z; // assign distance
    Propagate_in_environment_1;
    Plot_1;
    Plot_1.FileID++; // increment plot ID
    z += 1000 mm; // increment z by 1 mm
} // end of loop over z
until ( z> 1 cm )
}@
```

次のリスティングはwhile-statement(underlineをしている)のboolean expressionの内部で&&-operator(logical AND)を使う場合を示す。

```
{
// example for metric units
// propagate a beam from z=5 mm to 10 mm in 1 mm steps
// save a plot for each intermediary step
    Beam_Source_1;
    SaveDistribution_1;
    Plot_1.FileID - 20; // plot Ids shall start at 20
    z = 5 mm; // assign start distance to z
while ( ( z<= 1 cm ) && ( Plot_1.FileID <= 99 ) )
{ // begin of the loop over z
    Read_Distribution_1;
    Propagate_in_environment_1.distance = z; // assign distance
    Propagate_in_environment_1;
    Plot_1;
    Plot_1.FileID++; // increment plot ID
    z += 1000 um; // increment z by 1 mm
} // end of loop over z
}@
```

拡張されたBoolean expressionはloop内に用いたif-break lineを廃れたものとすることを留意されたい。したがってそれは関数性がないものとして削除してよい。